Self-Supervised Representation Learning for Census-Independent Population Estimation Using High-Resolution Satellite Imagery

Marina Kazamia

4th Year Project Report
Artificial Intelligence and Computer Science
School of Informatics
University of Edinburgh

2024

Abstract

Precise population maps are crucial for disaster response, resource allocation, and urban planning. While national censuses provide comprehensive population data, they are laborious, costly, and infrequent. During the intercensal period, population can change significantly due to factors such as rapid migration, natural disasters, and conflicts. The Sustainable Census-Independent Population Estimation (SCIPE) method, which uses high-resolution satellite imagery and a microcensus as a response variable, has shown promise but requires improvement in representation learning. This study extends SCIPE to study how self-supervised pretraining could help SCIPE produce better population estimates for two districts in Mozambique. We evaluated 25 pretraining configurations, including using several learning signals, dataset domains, sample sizes, and backbone architectures. Our findings indicate that finetuned contrastive models, specifically SimCLR, pretrained on very-high-resolution remote sensing images achieve superior per-tile population estimations. The TOV model, combining ImageNet and satellite images for self-supervised pretraining, achieved the lowest aggregate percentage error (AggPE = 0.6%). Additionally, ViT-B models pretrained on natural images outperformed ResNet-50 models, with a supervised ViT-B model outperforming SCIPE on all evaluation metrics. Our research establishes the superiority of self-supervised models pretrained on high-resolution remote sensing images over general ImageNet pretraining, marking a significant advancement in census-independent population estimation. Overall, this study enhances SCIPE's sustainability by using label-free datasets and avoiding manual annotation of zero-population tiles, making it a robust tool for population mapping.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Marina Kazamia*)

Acknowledgements

I would like to thank my supervisor, Dr. Sohan Seth, for his guidance and trust over the past year;

Karthik Mohan and Matthew Wisniewski for generously offering their time and insight with me on the SCIPE codebase;

Chloe Downing for her help as my student advisor throughout this challenging year;

my boyfriend, family, and friends for the invaluable support they have given me.

The work presented here wouldn't have been possible without all these people.

Table of Contents

Chapter 1

Introduction

1.1 Motivation

Precise population maps are essential for informed decision-making in various public interest areas, such as disaster response, resource allocation, urban planning, and tracking progress toward the Sustainable Development Goals (Neal et al., 2022). The national population and housing census is the most comprehensive and reliable data source for this purpose. A census is typically collected every ten years in most countries. However, censuses are laborious and expensive to build, which is why the intercensal period can sometimes extend up to several decades (Robinson et al., 2017). Recently, there have been several attempts to produce high-resolution population maps, for example over a 100 m grid (Linard et al. 2012; Tiecke et al. 2017; Bondarenko et al. 2020; Metzger et al. 2022; Neal et al. 2022). Satellite images are the main data used by the above approaches while depending on their use of census data as a dependent variable, they can be separated into census-dependent methods and census-independent methods.

Census-dependent population estimation (also known as population disaggregation or top-down estimation) uses the latest census data available to train a model to estimate a population density by projecting coarse-resolution data across a finer-resolution grid. Census-independent population (or bottom-up estimation) on the other hand, relies on surveys (or microcensuses) to train a predictive model that can provide population estimates at non-surveyed grid tiles (Metzger et al., 2022). Census-independent methods can improve the spatial and temporal resolution of census–dependent approaches, however, most of the existing census-independent approaches require hand-crafted features which are very laborious, time-consuming and expensive . Moreover, features can vary significantly between publications, making them less transferable to other geographic areas and countries and less sustainable for reusability (Neal et al., 2022).

For this reason, Neal et al. (2022) propose the Sustainable Census-Independent Population Estimation (SCIPE) method. SICPE uses representation learning for top-down population estimation from very-high-resolution (0.5 m) satellite imagery using a microcensus in two districts of Mozambique. They aggregated household survey data into a 100m grid to generate population counts, creating labeled grid tiles. Features are then

extracted using a pretrained Barlow Twins model (Zbontar et al., 2021) on ImageNet (Krizhevsky et al., 2012). This method successfully generated medium-resolution (100 m) population density maps without the need for manual annotation of satellite images and with minimal computational resources. Although SCIPE did not outperform estimations based on manually-extracted building footprint features on every evaluation metric, it remains a sustainable and transferable approach. Thus, a gap remains to be filled in extracting meaningful representations for top-down population estimation using satellite (or remote sensing) images and deep learning methods.

The performance of deep neural networks relies significantly on the size and quality of the training data (Wang et al., 2022c). Thus, numerous large-scale annotated datasets have been created for supervised training over the past decade, to help through transfer learning to train tasks where task-specific datasets are small and training labels are scarce (Krizhevsky et al. 2012; Zhou et al. 2017; Krizhevsky et al. 2009; Oquab et al. 2023). Models trained on ImageNet specifically, have been driving advancements in several fields, by using them as pretrained models and finetuning them for other tasks. However, annotating large-scale datasets is an extremely labor-intensive, time-consuming, and costly process.

Moreover, many recent studies have shown that general ImageNet pretraining may not be sufficient to learn useful representations for downstream remote sensing image understanding (RSIU) tasks (Wang et al. 2022a; Tao et al. 2023b; Huang et al. 2024; Wang et al. 2023). Wang et al. (2022c) argue that while transfer learning is very useful for datasets with limited labeled samples, the supervised pretraining learns useful representations for tasks of similar domain but it does not perform as well on tasks that are of different domain. There are several domain gaps between remote sensing (RS) and natural images arising from differences in capture perspectives, image resolutions, and object appearances, ultimately hindering the performance of RS image understanding (Huang et al., 2024). This find has led to many efforts in building large-scale annotated remote sensing datasets (Long et al. 2021; Wang et al. 2023).

However, creating a substantial remote sensing dataset is highly challenging due to the tedious and expertise-driven nature of accurately annotating remote sensing images (RSIs). Additionally, annotation methods vary significantly depending on the task; for instance, scene classification requires image-level annotation, whereas semantic segmentation requires pixel-level annotation (Tao et al., 2023a).

Furthermore, relying solely on manual labels as supervised signals is insufficient for training a good vision model. Manual labels function only as external guidance for fitting a model to the training data. In contrast, the intrinsic information embedded within the massive amounts of remote sensing data is theoretically richer and more fundamental than the semantic information provided by human-labeled samples. Consequently, human-labeled samples may fall short in annotating complex, multi-object scenes with multiple or ambiguous semantic meanings, leading to limited feature representation learning (Tao et al., 2023b).

In contrast, acquiring large amounts of unlabeled RSIs is relatively easy due to the increasing number of satellites dedicated to Earth observation. This abundance of unlabeled images, coupled with the limited availability of labeled data, makes representation

learning on RS images particularly suitable for self-supervised learning (SSL). SSL methods utilize vast amounts of unlabeled data by exploiting the inherent structure within the data to generate supervised signals (Jing and Tian, 2020). The ability to train deep learning models without the need for human-annotated labels, along with the remarkable success of SSL methods on natural images (Chen et al. 2020a; Chen et al. 2020b; Grill et al. 2020; Caron et al. 2021) has led to significant interest in applying SSL to RS images.

In this study, we use the SCIPE pipeline to explore the effect of different self-supervised pretraining paradigms on the downstream task of population estimation. We investigate the impact of multiple factors, including different learning signals, such as supervised versus self-supervised methods and contrastive [2.2.3] versus masked image modeling (generative) approaches [2.2.1]. Additionally, we analyze dataset characteristics, particularly focusing on the domain of images (natural images versus remote sensing images), sample size, and spatial resolution. Finally, we evaluate backbone network characteristics, comparing architectures like ResNet-50 [2.3.2.1], Swin [2.3.2.3], and ViT [2.3.2.2], along with network capacity.

1.2 Objective

The objective of this project is to study how self-supervised pretraining could help SCIPE produce better population estimates. Thus, we conduct ablation-like experiments using state-of-the-art self-supervised learning algorithms to assess the effects of different pretraining paradigms on our downstream task.

To the best of our knowledge, we are the first to implement a self-supervised learning pipeline for census-independent population estimation. We would like this work to be the first stepping stone of future research in the area.

1.3 Contributions

The contributions of this project are the following:

- We extend the SCIPE pipeline to incorporate self-supervised pretraining and the use of vision transformer models.
- We provide a comprehensive study of how different pretraining paradigms affect SCIPE population estimation, including pretraining with different learning signals, data domains, image resolutions, size of dataset, backbone architectures, and model capacity.
- We improve the performance of SCIPE across all evaluation metrics, avoiding the use of manually-constructed zero-population tiles, or the use of any zero-population tiles altogether.
- We establish the superiority of self-supervised models pretrained on high-resolution remote sensing images over the general ImageNet pretraining for learning good representations.

- We present the first self-supervised population estimation method with a 0.06% aggregate percentage error.

1.4 Thesis Structure

The thesis is divided into five chapters:

Chapter 1 explains the motivation behind this project, and shares its objectives and contributions.

Chapter 2 discusses background research on deep learning for intercensal population estimation, self-supervised learning, and model pretraining for representation learning in remote sensing.

Chapter 3 introduces the natural image and remote sensing datasets used for our pretrained models. It then describes the self-supervised learning models employed for pretraining. Following that, it presents the SCIPE pipeline, along with the satellite imagery and microcensus data used for population estimation. Finally, it explains the evaluation metrics and the cross-validation techniques used to compare our models.

Chapter 4 presents the experiments undertaken and the results. It discusses the effect of learning signals, dataset characteristics, and backbone network characteristics on model performance.

Chapter 5 summarises the findings of this project, discusses limitations, and proposes areas of future work.

Chapter 2

Background

2.1 Deep Learning for Intercensal Population Estimation

Representation learning is a method through which a model can learn useful representations or features without human curation. Learned representations often result in much better performance than can be obtained with hand-designed representations (Goodfellow et al., 2016). The learned representations can then be used for transfer learning. Transfer learning is useful when we have limited labeled data for the task that we are working on. By pretraining your model to a large, well-established labeled dataset, like ImageNet (Krizhevsky et al., 2012), it allows you to get a better performance on datasets with limited labels by finetuning on existing labeled data with minimal computational resources.

The first to experiment with representation and transfer learning on very-high-resolution satellite images for the downstream task of population estimation has been Neal et al. (2022) in their approach Sustainable Census-Independent Population Estimation (SCIPE).

2.1.1 The SCIPE Pipeline and Code

SCIPE explores the use of representation learning in census-independent population estimation from very-high-resolution (50 cm spatial resolution) satellite imagery using a microcensus (survey data) in two districts of Mozambique. They have aggregated household survey data to a 100m grid to generate population counts producing labeled grid tiles. A Barlow Twins model (Zbontar et al., 2021) with a ResNet-50 (He et al., 2016) convolutional neural network architecture was pretrained on ImageNet to extract representations. Once these representations are obtained, the final layers of the neural network are replaced with a simple linear regressor head and the model is finetuned using satellite images of surveyed grid tiles. After this finetuning stage, the regression head is removed to obtain a feature representation vector for each tile. These feature vectors are then fed into a Random Forest regressor to make population predictions for each tile.

This approach was able to produce medium-resolution (100 m) population density

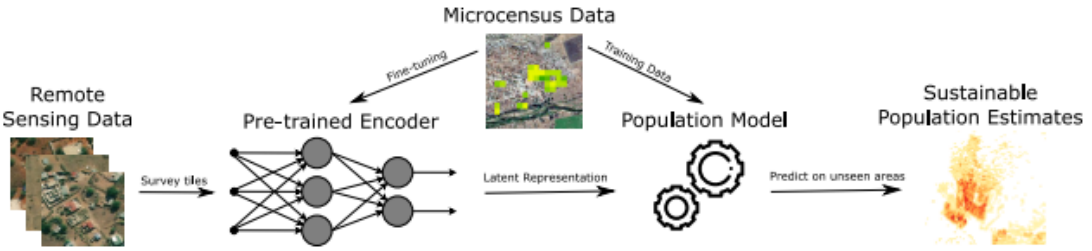

Figure 2.1: The SCIPE pipeline. Taken from Neal et al. (2022).

maps while avoiding manual annotation of satellite images (except for zero-population tiles) and only requiring minimal computational resources. Although SCIPE did not outperform building footprint area-based estimations, it is sustainable and transferable. An overview of their approach is shown in Figure 2.1.

This project is built on top of SCIPE; using the same pipeline and code. We adapt the code to incorporate self-supervised pretraining models of different backbone architectures, and different pretraining datasets into the pipeline. Other components of the pipeline, such as the code for data preprocessing, random forest regression, or model evaluation, have been available to us. The SCIPE codebase is written in Python, with the deep learning components utilizing PyTorch and FastAI.

2.2 Self-Supervised Learning

Self-supervised learning (SSL) is a novel approach of deep learning where the network attempts to learn representations using human-designed, task-agnostic learning signals to generate targets or pseudolabels for massive unlabeled data (Jing and Tian, 2020). These self-generated targets can then be used for pretraining in the same way as in supervised pretraining. Many recent SSL methods have demonstrated remarkable results on natural images (Chen et al. 2020b; Oquab et al. 2023; Grill et al. 2020; He et al. 2022).

SSL methods can be divided into three categories according to their choice of learning signal: Generative, Predictive and Contrastive.

2.2.1 Generative Learning Signals

Generative learning signals involve training a model to reconstruct the original input from a partially corrupted version. This method operates on the assumption that if the model can accurately recover the missing information, it has effectively learned the contextual features. First, the original image, x is corrupted by adding random noise, masks, or downsampling, resulting in a degraded version $\tilde{x}$. Then, a model $f(\cdot)$ with an encoder-decoder architecture learns features by minimizing the objective function $\|f(\tilde{x}) - x\|_2^2$ (Wang et al., 2022c). The masked autoencoder (MAE) (He et al., 2022) is one such approach, which masks a high portion ($\sim 75\%$) of the input image with random patches and reconstructs the missing patches in the pixel space. MAE learns high-capacity models that generalize well in downstream tasks.

2.2.2 Predictive Learning Signals

Predictive learning signals focus on learning semantic context features rather than dealing with pixel-level details like generative learning signals. Instead, these methods focus on predicting specific properties of the data using pretext tasks. These tasks involve designing a suitable challenge for the dataset, generating self-labels, and training a model to predict these labels to learn data representations (Jing and Tian, 2020). Pretext tasks utilize various context information of the input images and can be categorized based on context attributes into spatial context and spectral context. Spatial context prediction includes the pretext tasks of relative position prediction (Doersch et al., 2015), solving jigsaw puzzles (Noroozi and Favaro, 2016), and image rotation prediction (Gidaris et al., 2018), whereas spectral context prediction is achieved through colorization tasks (Zhang et al., 2016).

2.2.3 Contrastive Learning Signals

The effectiveness of predictive SSL is largely dependent on well-designed pretext tasks, which can be difficult to create and may lead to task-specific representations, reducing the model's generalizability. Contrastive methods overcome this limitation by giving the network flexibility to learn high-level representations without depending on a single pretext task. Extensive experimental research in psychology indicates that infants primarily learn perceptual categories through observation, not linguistic supervision. This process allows them to recognize different versions of the same object (invariance) and differentiate objects by their appearances (distinguishability). Contrastive learning signals are designed to imitate this by bringing augmented views of the same image closer together and separating views of different images to learn invariant and distinguishable visual features (Tao et al., 2023a). However, enforcing only similarity between pairs can result in a trivial solution known as model collapse, where the model maps all inputs to the same representation (Caron et al., 2021). To prevent this, various strategies have been proposed, creating a subtaxonomy of contrastive SSL methods that includes negative sampling (Chen et al. 2020a; Chen et al. 2020b), clustering (Caron et al., 2018), knowledge distillation (Grill et al. 2020; Caron et al. 2021), and redundancy reduction (Zbontar et al., 2021).

2.3 Model Pretraining for Remote Sensing Tasks

Pretraining models is crucial for enhancing the performance of deep neural networks on remote sensing images (RSIs). Historically, most approaches have relied on the ImageNet dataset (Krizhevsky et al., 2012) for pretraining, resulting in improved performance in classification, segmentation, and detection tasks through task-specific designs (Tong et al. 2020; Zheng et al. 2020; Ding et al. 2019). However, significant domain gaps exist when transferring these pretrained models from natural images to remote sensing (RS) tasks due to the substantial differences between these image types (Huang et al., 2024).

Recently, a new approach has been proposed that involves pretraining deep models on a large-scale RS dataset called MillionAID (Long et al., 2021) in a fully supervised

manner (Wang et al., 2022a). This method has shown that pretraining on RS datasets can enhance the performance of both convolutional neural networks (CNNs) and vision transformers (ViTs) (Wang et al., 2022b). Nonetheless, the need for extensive labeled data poses a challenge for pretraining larger models, since annotating RS datasets is very challenging and costly due to the expertise needed to undertake the task. Moreover, label noise is another issue in remote sensing since satellite images are complex and used for several different tasks, so the perfect label does not exist. The impressive performance of SSL in natural image representation learning, together with the issues of label noise and label scarcity in remote sensing datasets has encouraged many studies on the potential of SSL in the remote sensing community (Wang et al. 2022c; Tao et al. 2023a).

2.3.1 Self-Supervised Pretraining for Representation Learning in Remote Sensing

Initially, contrastive learning (CL) was the most popular SSL method within the RS community, with much work being done in integrating RS characteristics into CL design to encourage more specialized representation learning from RS images. For instance, Jean et al. (2019) proposed Tile2Vec, an unsupervised representation learning method that assumes geographically proximate tiles exhibit semantic similarity, using metric learning for unsupervised tile learning. Geography-aware MoCo (Ayush et al., 2021) bridged the gap between self-supervised and supervised learning on various RS downstream tasks by using spatially aligned images over time to form temporal positive pairs and employing a geo-location pretext task during training to enhance representation learning of RS images. Similarly, SeCo (Manas et al., 2021) used images from the same location at different times as positive pairs, providing representations with time-varying and invariant features.

Recently, the masked image modeling (MIM) method has gained popularity in the RS community due to the success of Vision Transformers (ViT) in various RS downstream tasks. RingMo (Sun et al., 2022) applied the SimMIM method (Xie et al., 2022), introducing a new masking strategy for self-supervised representation learning on a dataset of 3 million unlabeled RS images. Fine-tuning results on various downstream tasks showed that this new masking strategy was more suitable for RS images, and the representations learned by RingMo generalized well to various RS downstream tasks. MAE (He et al., 2022) was also used to pretrain large vision models customized for RS, with results demonstrating the effectiveness of MIM pretraining (Wang et al., 2022b).

The above SSL methods for remote sensing are restricted to using either contrastive learning or masked image modeling. Muhtar et al. (2023) proposes contrastive mask image distillation (CMID) in an effort to improve the generalization performance of remote sensing pretraining. Muhtar et al. (2023) argue that because CL focuses on inter-image semantic relationships and overlooks intra-image structure it fails to perceive semantic information at different spatial locations within images, resulting in poor performance in dense prediction tasks such as object detection and semantic segmentation. Conversely, MIM learns intra-image structure and captures the contextual information of each pixel within an image, resulting in representations with local spatial

perceptibility at the expense of global semantic separability. Consequently, MIM pretrained models excel in dense prediction tasks but are less effective in classification tasks.

More studies approach this inability of RS contrastive learning in capturing low- to mid-level features by pretraining models using both RS and natural images (Huang et al. 2024; Tao et al. 2023b; Risojević and Stojnić 2021; Zhang et al. 2022). Tao et al. (2023b) introduces TOV which freezes the shallow and middle layers of a natural image SSL pretrained model before training on the RS dataset to avoid catastrophic forgetting of general knowledge from the natural images while adapting to RS images. Huang et al. (2024) propose GeRSP, a knowledge distillation method that simultaneously trains a supervised network with natural images and a self-supervised network with RS images. These approaches generalize better to object detection and segmentation tasks, indicating that pretraining on natural images provides useful general knowledge representations.

Moreover, the spatial resolution of the RS pretraining dataset can too determine the spatial information that the model extracts. Tao et al. (2023a) argue that a pretraining dataset with high spatial resolution is critical for good self-supervised feature learning. So much so, that having high-resolution pretraining data might compensate for having a large domain gap between pretraining and downstream datasets. Chopra et al. (2023) proposes a domain adaptation-based self-supervised representation learning approach for classifying satellite images. The model gets pretrained on one source dataset and evaluated on a different target dataset. Using three very-high-resolution to high-resolution (0.1 - 10 m) datasets, the proposed method has surpassed existing results by 1% with less training data.

2.3.2 Backbone Network Architectures

The most commonly used network architecture for computer vision (CV) tasks (including remote sensing image understanding) has been convolutional neural networks (especially ResNets (He et al., 2016)), but given the recent progress of natural language processing (NLP) with transformers (Vaswani et al., 2017), many studies have arisen in exploring the adoption of transformers to CV (Dosovitskiy et al. 2020; Caron et al. 2021; Oquab et al. 2023; Liu et al. 2021; Xu et al. 2021; Zhang et al. 2023) and remote sensing (Wang et al. 2022a; Wang et al. 2022b; Muhtar et al. 2023).

Here, we introduce briefly the different backbone architectures that are used in this study.

2.3.2.1 ResNet-50

ResNet50 (He et al., 2016) is a deep convolutional neural network (CNN) with 50 layers, organized into ”bottleneck” residual blocks. Residual blocks address the vanishing/exploding gradient problem in deep neural networks by incorporating skip connections. These skip connections allow gradients to bypass certain layers. This helps prevent the gradients from becoming too small as they propagate back through the network, facilitating effective training. Moreover, the bottleneck structure of the blocks

reduces the computational load of training a ResNet model through parameter reduction. It involves a three-layer sequence: a 1x1 convolution to reduce the dimensionality of the feature maps, followed by a 3x3 convolution, and another 1x1 convolution to restore the dimensionality. After the residual blocks, a global average pooling layer reduces the spatial dimensions.

2.3.2.2 ViT – Visual Transformer

ViT (Dosovitskiy et al., 2020) divides an input image into fixed-size non-overlapping patches, flattens each into a 1D vector, and projects these vectors into a lower-dimensional space to create patch embeddings. To retain spatial information, position embeddings are added to the patch embeddings. The model uses a Transformer encoder (Vaswani et al., 2017) with multiple layers of multi-head self-attention (MHSA) mechanisms and feed-forward neural networks to process these embeddings. A special [CLS] token is prepended to the sequence, and its final representation is used for classification tasks. The self-attention mechanism enables the model to capture long-range dependencies and contextual relationships between patches.

2.3.2.3 SwinT – Shifted Window Transformer

The Swin Transformer (Liu et al., 2021) is another neural network architecture tailored for CV tasks. It first divides images into fixed-size non-overlapping patches, and creates patch embeddings in the same way as Vision Transformers. These embeddings are further divided into non-overlapping windows of fixed size and self-attention is computed locally within each window. Each window produces a local feature representation, capturing the semantics within the confined space of the window. To capture global context, the windows are shifted between layers. Finally, neighboring windows are merged to form larger windows, reducing the number of tokens. This is done by concatenating the features of adjacent patches and applying a linear transformation. This stage acts similarly to pooling layers in CNNs, progressively reducing the spatial dimension while maintaining knowledge about both local and global dependencies.

Chapter 3

Methodology and Data

In this chapter, we first introduce the natural image and remote sensing datasets that have been used for our pretrained models. Then, we provide details for the self-supervised learning models that are used independently for pretraining. Next, we present the SCIPE pipeline together with the satellite imagery and microcensus that are used for the task of population estimation. Lastly, we show what evaluation metrics and how cross-validation have been used to compare our models.

3.1 Datasets Used in Pretraining

Here we present the datasets that have independently been used for the pretraining of our models. Table 3.1 offers a summary of the properties of the different datasets.

3.1.1 SIRI-WHU

The SIRI-WHU dataset[1] contains 2400 images categorized into 12 classes. Sourced from Google Earth, this dataset primarily features metropolitan areas in China and was compiled by Wuhan University's RS IDEA Group. The 12 classes include Agriculture, Commercial, Harbor, Idle Land, Industrial, Meadow, Overpass, Park, Pond, Residential, River, and Water, with each class consisting of 200 images. These images are 200 x 200 pixels in size and have a spatial resolution of 2 meters.

3.1.2 UC Merced

The UC Merced dataset[2] (Yang and Newsam, 2010) consists of images manually extracted from large-sized images within the United States Geological Survey (USGS) National Map Urban Area Imagery collection, covering various cities across the United States. This extensive ground truth dataset includes 21 land-use types, each represented by 100 images. The 21 classes are agricultural, airplane, baseball diamond, beach, buildings, chaparral, dense residential, forest, freeway, golf course, harbor, intersection,

[1] http://www.lmars.whu.edu.cn/prof_web/zhongyanfei/e-code.html

[2] http://weegee.vision.ucmerced.edu/datasets/landuse.html

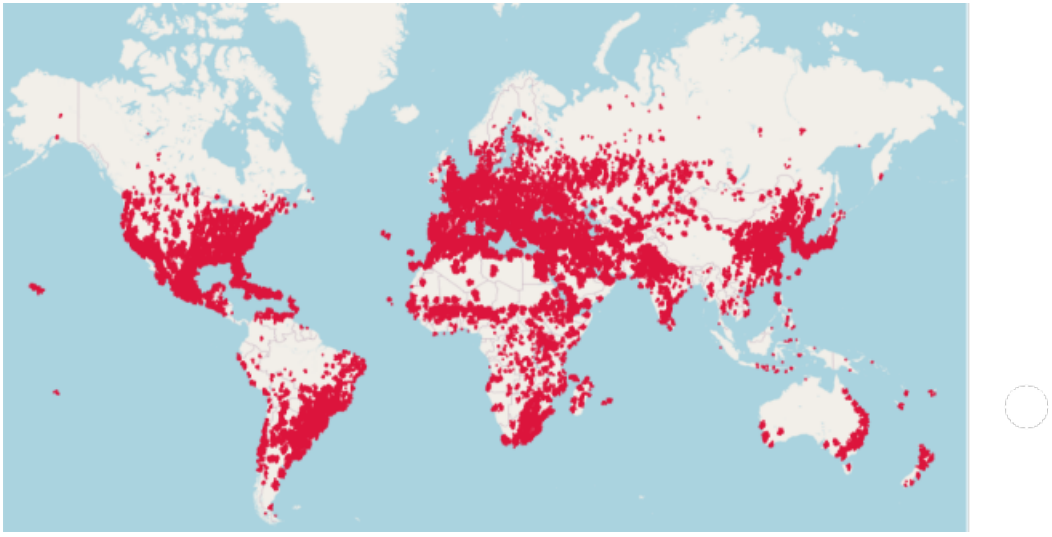

Figure 3.1: The geographical distribution of the SSL4EO-S12 dataset. Taken from Wang et al. (2023).

medium residential, mobile home park, overpass, parking lot, river, runway, sparse residential, storage tanks, and tennis court. Each image in this collection has a resolution of 0.3 meters and measures 256 x 256 pixels.

3.1.3 MLRSNet

The MLRSNet dataset[3] (Qi et al., 2020) provides various satellite-based perspectives from around the world, featuring high-resolution optical satellite images. It includes 109,161 remote sensing photographs, categorized into 46 classes, with each category containing between 1,500 and 3,000 example images. The photos, each sized at 256 x 256 pixels, have spatial resolutions ranging from 10 meters to 0.1 meters. This dataset is suitable for tasks such as image segmentation, retrieval, and multi-label classification.

3.1.4 SSL4EO-S12

SSL4EO-S12 (Wang et al., 2023) is remote sensing dataset of medium resolution that provides global geospatial coverage as seen in Figure 3.1. The complete SSL4EO-S12 dataset contains 3,000,000 Sentinel-1 dual-pol SAR, Sentinel-2 top-of-atmosphere (level-2C) multispectral, and Sentinel-2 surface reflectance (level-2A) multispectral triplets over four seasonal timestamps.

Since the images in the dataset that we will be using for population estimation consist of only the 3 bands of RGB, the MoCo-v2 model that was acquired from (Wang et al., 2023) is only trained on a subset of the SSL4EO-S12 dataset, i.e., the satellite images that are of RGB bands. These images are part of the Sentinel-2 level-2C only, have a patch size of 264 x 264 pixels, and an image resolution of 10m. Unfortunately, the size of this subset is not mentioned by the authors.

[3] https://data.mendeley.com/datasets/7j9bv9vwsx/2

Dataset	Number of images	Image size	Spatial Resolution	Geographic Range
Remote Sensing Images				
SIRI-WHU	2,400	200x200	2 m	China
UC Merced	2,100	256x256	0.3 m	United States
MLRSNet	109,161	256x256	0.1 - 10 m	Worldwide
SSL4EO-S12 (RGB)	$<$ 3,000,000	264x264	10 m	Worldwide
Million-AID	1,000,848	110x110 - 31,672x31,672	0.5 - 153 m	Worldwide
Potsdam	21,888	256x256	0.5 m	Urban Germany
TOV-RS	500,000	600x600	1 - 20 m	Worldwide
Natural Images				
ImageNet	1,281,167	224x224	–	–
TOV-NI	1,000,000	224x224	–	–
LVD-142M	142,000,000	518x518	–	–

Table 3.1: Characteristics of the datasets used for pretraining.

3.1.5 Million-AID

The Million-AID dataset (Long et al., 2021) consists of 1,000,848 images with 51 scene categories. It's primary purpose was to serve as a scene classification dataset. Million-AID was collected from Google Earth. It features images of a vast range of resolutions and sizes, ranging from 0.5 to 153 m per pixel and 110 to 31,672 pixels per image, respectively.

3.1.6 Potsdam

The Potsdam dataset [4] includes 38 tiles, each measuring 6000 × 6000 pixels with a spatial resolution of 0.5 meters. The dataset is manually labeled into six categories: low vegetation, tree, building, impervious surface, car, and clutter. The pretrained models on Potsdam that we have used in this study utilize a slightly modified dataset. The 38 tiles have been cropped into 256 × 256 pixel patches with a stride of 128 pixels, resulting in a total of 21,888 images for pretraining.

3.1.7 TOV-NI and TOV-RS

Tao et al. (2023b) make available a natural image dataset, TOV-NI, and a remote sensing dataset, TOV-RS. TOV-NI includes 1 million web-crawled images representing a wide range of visual concepts. The images are selected to ensure high diversity, capturing different categories, perspectives, and lighting conditions. TOV-RS consists of 0.5 million class-balanced samples, collected using automated sampling methods guided

[4] https://www.isprs.org/education/benchmarks/UrbanSemLab/2d-sem-label-potsdam.aspx

by geographic data products. This ensures that the images represent a wide range of geographic regions and land cover types. TOV-RS has a spatial resolution between 1 to 20 metres per pixel, and each image has a size of 600 x 600.

3.1.8 ImageNet-1K

The ImageNet-1K dataset (Krizhevsky et al., 2012) is a benchmark dataset in computer vision research, specifically in the image classification domain. The dataset contains 1,281,167 training images, distributed across 1,000 distinct classes. Each class represents a different object, animal, scene or concept. The images vary in size, resolution and quality, but they are typically resized to 224x224 pixels for training and evaluation purposes.

3.1.9 LVD-142M

Oquab et al. (2023) have created the LVD-142M dataset by selecting images from a vast pool of uncurated and curated data. The curated datasets, include the training split of ImageNet-1k, ImageNet-22k, Google Landmarks, and various fine-grained datasets. For the uncurated data, images from a publicly available repository of crawled web data have been collected, resulting in 1.2 billion unique images after filtering for safety and removing restricted URLs. To enhance diversity, near-duplicate images from the uncurated data have been removed, through a self-supervised image retrieval pipeline. The remaining 142M images make up the LVD-142M dataset.

3.2 Self-Supervised Pretraining

In this section, we describe the different self-supervised algorithms that we have experimented with for pretraining, and provide implementation details for the pretrained models that we have used.

3.2.1 SimCLR

SimCLR (Chen et al., 2020a), or Simple Framework for Contrastive Learning of Visual Representations, is a model that learns through a contrastive learning signal [2.2.3]. The core idea of SimCLR is to maximize the similarity between differently augmented views of the same image while minimizing the similarity between views of different images. This method consists of four main components: data augmentations, a backbone encoder network (typically a ResNet-50), a projection head, and a contrastive loss function.

For each given input image, random augmentations are applied generating two views. The combination of random cropping and resizing, color distortions, and Gaussian blur have been found to yield the best performance by the authors. Next, the encoder extracts feature vectors from the augmented images. The output of the encoder then goes through the projector, a multilayer perceptron (MLP) with one hidden layer, to

map the representation vectors into a lower-dimensional space where the contrastive loss is applied.

The loss function used here is called NT-Xent (normalized temperature-scaled cross entropy loss). A minibatch of N examples is randomly selected. The contrastive prediction task is defined using pairs of augmented examples derived from this minibatch, resulting in $2N$ data points. Negative examples are not sampled explicitly; instead, for each positive pair, the other $2(N-1)$ augmented examples within the minibatch are considered as negative examples. Let $\text{sim}(u,v) = \frac{u^\top v}{\|u\|\|v\|}$ denote the cosine similarity of u and v. Then the NT-Xent loss function for a positive pair of examples (i, j) is defined as:

$$L_{i,j} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{I}_{[k\neq i]} \exp(\text{sim}(z_i, z_k)/\tau)}, \tag{3.1}$$

where $\mathbb{I}_{[k\neq i]} \in \{0,1\}$ is an indicator function that evaluates to 1 if $k \neq i$, and τ denotes a temperature parameter. The final loss is computed across all positive pairs, both (i, j) and (j,i), in a minibatch. Finally, to keep it simple, instead of using a memory bank, SimCLR employs a large batch size to increase the number of negative examples per positive pair.

All these components contribute to SimCLR's ability to improve the quality of the learned representations, and thus, experiments with SimCLR have achieved very good results. Chen et al. (2020a) show that a linear classifier trained on the self-supervised representations learned by SimCLR achieves 76.5% top-1 accuracy on ImageNet, matching the performance of a supervised ResNet-50. Moreover, when finetuned with only 1% of the labeled data, SimCLR achieves 85.8% top-5 accuracy, significantly outperforming other self-supervised methods.

We use the SimCLR ResNet-50 model pretrained on ImageNet Chen et al. (2020a) to study the ability of this pretraining paradigm to generalize to our downstream task of population estimation using remote sensing images. The model was ran for a total of 100 epochs with a batch size of 4096. The temperature parameter of the loss function was set to 0.1. A LARS optimizer (You et al., 2017) is employed with a learning rate of 4.8 ($0.3 \times BatchSize/256$) and weight decay of 1×10^{-6}. The linear warm-up takes place for the first 10 epochs. A cosine decay scheduler is used then on the learning rate (Loshchilov and Hutter, 2016).

In addition, Chopra et al. (2023) have made three SimCLR models (with a ResNet-50 backbone), each trained on a different remote sensing dataset (MLRSNet [3.1.3], UC Merced [3.1.2] and SIRI-WHU [3.1.1]), publicly available. These models have been used for pretraining in this work to evaluate how well SimCLR can learn and transfer representations from remote sensing data. All three models follow the same training setting. The augmentations used are Gaussian blur, flipping, greyscale, rotation, and resizing. The training takes place over 100 epochs with a batch size of 256. A Stochastic Gradient Descent (SGD) optimizer is employed with a momentum of 0.9. The learning rate and weight decay are both set to 0.0005.

3.2.2 MoCo-v2 – Momentum Contrast

The MoCo-v2 model (Chen et al., 2020b) is an improved version of the original MoCo (Momentum Contrast) framework (He et al., 2020) for self-supervised visual representation learning. The MoCo architecture is shown in Figure 3.2. It takes two inputs, a query and a set of keys. A query is an image taken from the dataset, and a key is an augmented version of the query. MoCo consists of an encoder and a momentum encoder. Similarly to SimCLR, an encoder can be any convolutional neural network architecture, but a ResNet50 (He et al., 2016) network is commonly used to extract feature representations.

Due to the large size of the feature queue, it is impractical to update the key encoder using backpropagation. Therefore, MoCo uses a momentum update and so the parameters of the momentum encoder are updated using a moving average of the query encoder's parameters. This approach helps to stabilize the learning process and maintain consistency in the representations over time. The update equation is:

$$\theta_k \leftarrow m\theta_k + (1-m)\theta_q \tag{3.2}$$

where θ_k are the parameters of the momentum encoder, θ_q are the parameters of the query encoder, and m is the momentum coefficient (typically close to 1).

Another key component of MoCo is the feature queue. The feature queue is a large dictionary that acts as a queue storing encoded key representations. The queue is dynamic, meaning that it is constantly updated with new negative sample embeddings. Thus, MoCo can make use of a large number of negative samples without a memory bank or a large batch size, avoiding any high memory and computational costs.

Finally, the contrastive InfoNCE loss is used to train the encoder networks. Given an encoded query q and a set of encoded samples $\{k_0, k_1, k_2, \ldots\}$ which are the keys of a dictionary, there is a single key (denoted as k^+) in the dictionary that matches q. A contrastive loss function is designed to yield a low value when q is similar to its positive key k^+ and dissimilar to all other keys (considered negative keys for q). With similarity measured by dot product, InfoNCE loss is defined as follows:

$$L_q = -\log \frac{\exp(q \cdot k^+/\tau)}{\sum_{i=0}^{K} \exp(q \cdot k_i/\tau)} \tag{3.3}$$

where τ is a temperature hyper-parameter. The summation includes one positive and K negative samples.

MoCo-v2 incorporates two main architectural improvements from MoCo-v1. Both updates are inspired from the SimCLR model. Firstly, MoCo-v2 incorporates a MLP projection head. This projection head consists of two fully connected layers with a ReLU activation function in between. The hidden layer has 2048 dimensions, and the output layer projects the features into a 128-dimensional space. Secondly, MoCo-v2 applies stronger data augmentation techniques than in the original MoCo, just like SimCLR.

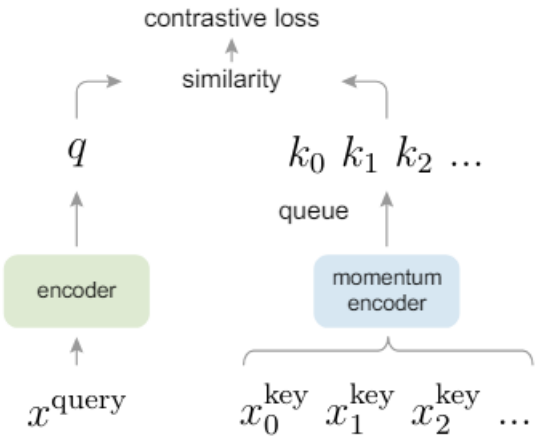

Figure 3.2: The Momentum Contrast (MoCo) architecture. Taken from He et al. (2020).

After 200 epochs of training with a batch size of 256, MoCo-v2 achieves a top-1 accuracy of 67.5% on ImageNet linear classification. This performance is 5.6% higher than SimCLR under the same conditions.

We use various MoCo-v2 pretrained models in this work. We present them below.

3.2.2.1 MoCo-v2 pretrained on ImageNet

He et al. (2020) released their MoCo-v2 (ResNet-50) model trained on ImageNet-1K. The training utilized the data augmentations of random resized cropping, color jittering, horizontal flipping, Gaussian blur, and grayscale conversion. The batch size was set to 256, and the optimizer was SGD with a momentum of 0.9. The initial learning rate was 0.03, following a cosine decay schedule without restarts, and the weight decay was 0.0001. The momentum encoder update coefficient was 0.999, and the model was trained for 800 epochs. A temperature parameter of 0.2 was used for the contrastive loss, and the queue size for negative samples was 65,536.

3.2.2.2 MoCo-v2 pretrained on SSL4EO-S12

Wang et al. (2023) introduce a MoCo-v2 model that has been trained on a subset of the SSL4EO-S12 dataset [3.1.4] . With a ResNet50 backbone, an SGD optimizer, a cosine learning rate scheduler with a learning rate of 0.03 and a batch size of 256, Moco-v2 has been trained for 100 epochs.

3.2.2.3 MoCo-v2 pretrained on Potsdam

Muhtar et al. (2023) make available a MoCo-v2 model trained on Potsdam [3.1.6]. They trained with a batch size of 64 for 400 epochs. All other training settings are identical to 3.2.2.1.

3.2.3 BYOL - Bootstrap Your Own Latent

Grill et al. (2020) introduced the model 'Bootstrap Your Own Latent' (BYOL); a self-supervised contrastive learning model which achieves state-of-the art performance without the use of negative samples. BYOL follows a *knowledge distillation* framework: given an augmented view of an image, an *online network* is trained to predict the *target's network* embedding of a different augmented view of the same image. A sequence of representations of increasing quality are built by iterating this procedure and using subsequent online networks as target networks as training continues.

The architecture between the online and target pipeline is asymmetric. The online network consists of an encoder, a projector and a predictor, whereas the target network consists of just an encoder and a projector. Similar to the concept of momentum updates in MoCo [3.2.2], the parameters, ξ, of the teacher network in BYOL are an exponential moving average of the online network's parameters, θ. This serves as means of stabilizing the bootstrap step to avoid collapsed (trivial) solutions.

Given an image x, BYOL produces two augmented views, v and v'. The same augmentations as SimCLR [3.2.1] are used. The online network processes v to give a representation $y_\theta = f_\theta(v)$ and a projection $z_\theta = g_\theta(y)$. The target network processes v' to give $y'_\xi = f_\xi(v')$ and $z'_\xi = g_\xi(y')$. Both projectors are multi-layer perceptrons (MLPs) that include a linear layer with an output size of 4096, followed by batch normalization, rectified linear units (ReLU) (Nair and Hinton, 2010), and a final linear layer with an output dimension of 256. Unlike SimCLR, the output of this MLP is not batch normalized. The online network then outputs a prediction z'_ξ using $q_\theta(z_\theta)$. The predictor q_θ uses the same architecture as the projector g_θ. Both $q_\theta(z_\theta)$ and z'_ξ are l_2-normalized. Finally, the mean squared error (MSE) loss is used between the normalized predictions and target projections as follows:

$$L_{\theta,\xi} = \|q_\theta(z_\theta) - z'_\xi\|_2^2.$$

To symmetrize the loss, the authors also feed v' to the online network and v to the target network to compute $L'_{\theta,\xi}$. During training, the combined loss, L_{BYOL}, is minimized:

$$L_{BYOL} = L_{\theta,\xi} + L'_{\theta,\xi}.$$

The BYOL method with a ResNet50 backbone network achieves better top-1 and top-5 accuracies under linear evaluation on ImageNet than both SimCLR and MoCo-v2. Moreover, it seems that BYOL learns more generic representations than SimCLR since BYOL performs better on other classification datasets, however, it still can't beat the supervised benchmark. Finally, BYOL has also been evaluated on other downstream tasks such as semantic segmentation where BYOL performs better than SimCLR, MoCo and a supervised pretrained model (Grill et al., 2020).

In this study, we evaluate the ability of BYOL to produce representations that could help in our task of census-independent population estimation. Muhtar et al. (2023) make available a BYOL model pretrained on Potsdam [3.1.6]. The particular model was trained from scratch for 400 epochs with a batch size of 64. The LARS optimizer

with a cosine learning rate scheduler was employed. The base learning rate was set to 0.2 and was scaled linearly with the batch size (LearningRate = 0.2 × BatchSize/256). Additionally, a weight decay of 1.5×10^{-6} was applied, excluding batch normalization parameters and biases from both LARS adaptation and weight decay. For the target network, the exponential moving average parameter τ starts at $\tau_{\text{base}} = 0.996$ and is gradually increased to 1 during training.

3.2.4 DINO

Caron et al. (2021) proposes a new algorithm called DINO, which largely took inspiration from BYOL (Section 3.2.3). Similar to BYOL, the parameters of the teacher network are updated with an exponential moving average of the parameters of the student network, i.e. a momentum encoder (He et al., 2020). The approach is the same as in Section 3.2.3 with some exceptions.

First, DINO does not make use of a predictor like BYOL does. Here, the architectures of the online and teacher networks are identical. Many SSL algorithms differ in the way they stabilize and avoid collapse. Popular options are through contrastive loss, a predictor, batch normalizations or clustering constraints. Instead, DINO performs centering and sharpening of the momentum teacher outputs.

Caron et al. (2021) show through experiments that while centering encourages collapse to the uniform distribution, it also prevents a single dimension to dominate. Sharpening has the opposite effect. When both are applied, their effects are balanced and collapse is prevented when a momentum encoder is present. Sharpening is achieved by using a low temperature value in the teacher softmax normalization, which we discuss later. Centering, on the other hand, can be seen as an addition of a bias term c to the teacher network, g_t: $g_t(x) \leftarrow g_t(x) + c$. The update of the center c is as follows:

$$c \leftarrow mc + (1-m)\frac{1}{B}\sum_{i=1}^{B} g_{\theta_t}(x_i),$$

where $m > 0$ is a rate parameter and B is the batch size. DINO shows robustness to batch size variations since EMA updates the center c in a way that incorporates information over multiple batches.

Second, DINO uses a standard cross-entropy loss instead of a mean squared error to measure the similarity of the softmax-normalized outputs of the two networks. Given an input image x, both networks produce probability distributions over K dimensions, denoted as P_s and P_t. The probability P is derived by normalizing the output of the network g using a softmax function. Specifically,

$$P_s(x)^{(i)} = \frac{\exp(g_{\theta_s}(x)^{(i)}/\tau_s)}{\sum_{k=1}^{K}\exp(g_{\theta_s}(x)^{(k)}/\tau_s)},$$

where $\tau > 0$ is a temperature parameter that controls the sharpness of the output distribution. A similar formula applies to P_t with temperature τ_t.

Finally, in addition to the data augmentations used by BYOL, DINO applies a multi-crop strategy (Caron et al., 2020). DINO uses multi-crop augmentations for retrieving image

representations of both local and global importance. From a given image, a set of V different views is generated. Each set contains two *global* views, x_1^g and x_2^g and several *local* views of lower resolution. All crops are then passed through the student network, but only the global views are passed through the teacher. This is believed to encourage "local-to-global" correspondences through minimizing an adapted cross-entropy loss with stochastic gradient descent in the following way:

$$\min_{\theta_s} \sum_{x \in \{x_1^g, x_2^g\}} \sum_{\substack{x' \in V \\ x' \neq x}} H(P_t(x), P_s(x')),$$

where $H(a,b) = -a \log b$, and P_t and P_s represent the output probability distributions of the teacher and student network, respectively. The above loss function can be used on any number of augmented views, even only two.

The DINO framework works effectively on both CNNs and vision transformers (ViTs) without the need to modify the architecture. The neural network g consists of a backbone f and a projection head h such that $h : g = h \circ f$. f can be either a ViT or a ResNet. The features used for downstream tasks are the output of the backbone f. The projection head is a 3-layer multi-layer perceptron (MLP) with a hidden dimension of 2048, followed by l_2 normalization and a weight-normalized fully connected layer with K dimensions, similar to the design from SwAV (Caron et al., 2020). Notably, unlike standard convolutional networks, ViT architectures do not use batch normalization (BN) by default. Caron et al. (2021) make the DINO model with a ViT backbone entirely BN-free by excluding BN in the projection heads.

Caron et al. (2021) evaluate the performance of DINO-ResNet50 and DINO-ViT using linear and k-NN classification on ImageNet. They find that DINO-ViT-S performs significantly better than DINO-ResNet50. However, DINO-ResNet50 still performs better than other popular CNN-based SSL methods like SimCLR, MoCo-v2, Barlow-Twins, and BYOL. DINO-ViT-B/8 and DINO-ViT-S/8 achieve the best overall top-1 accuracy across architectures on linear classification with 80.1% and on k-NN classification with 78.3%, respectively. Moreover, they observe that ViT self-supervised pretraining transfers better than features trained on supervision. Finetuned DINO-ViT-B/16 models surpass the performance of supervised pretraining features in all 7 downstream classification datasets except one.

We evaluate the ability of the self-supervised DINO-ResNet50 and DINO-ViT-B/16 models pretrained on ImageNet to transfer useful visual representations for the task of population estimation. Both models follow the same training hyperparameters. Training takes place over 100 epochs with a batch size of 1024. An AdamW optimizer is employed, with the learning rate linearly increasing with the batch size during the first 10 epochs (LearningRate = 0.0005 × BatchSize/256). After this warm-up period, the learning rate decays according to a cosine schedule. The weight decay also follows a cosine schedule, starting from 0.04 and increasing to 0.4. The temperature τ_s is set to 0.1, and τ_t is linearly ramped up from 0.04 to 0.07 during the first 30 epochs. Finally, in addition to multi-crop, the data augmentations used in BYOL are applied: color jittering, Gaussian blur and solarization.

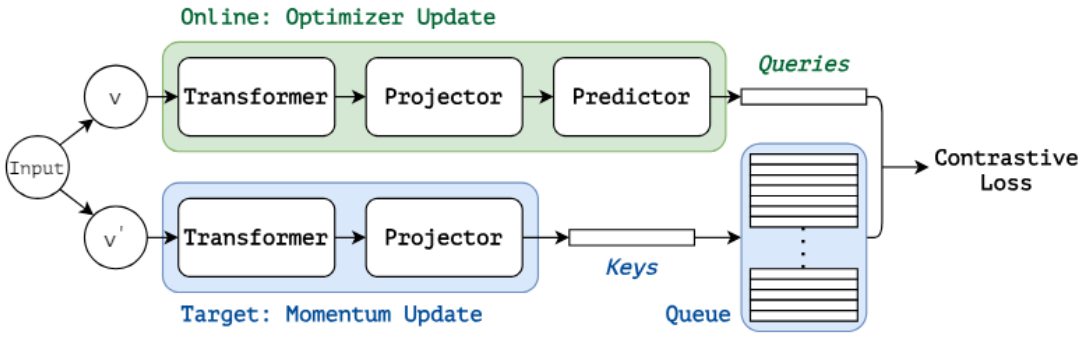

Figure 3.3: The pipeline of MoBy; a combination of popular SSL algorithms MoCo-v2 and BYOL. Taken from Xie et al. (2021).

3.2.5 MoBy

Xie et al. (2021) introduce MoBy, an approach to incorporate self-supervised learning with Swin transformer backbones. MoBy combines components of MoCo-v2 [3.2.2] and BYOL [3.2.3]. From Moco-v2 it inherits its momentum encoder, its key queue, and the contrastive loss. From BYOL, it inherits its asymmetric encoders, asymmetric data augmentations and its momentum scheduler.

The MoBy pipeline is illustrated in Figure 3.3. Both the online and target encoders consist of a Swin backbone and a 2-layer MLP projection head. The online encoder consists of an additional 2-layer MLP predictor. As also discussed in BYOL, the target encoder is updated through an exponential moving average of the online encoder's weights at each iteration. The online encoder is updated using a contrastive loss. More precisely, for an online view q, its contrastive loss is computed as

$$L_q = -\log \frac{\exp(q \cdot k_+/\tau)}{\sum_{i=0}^{K} \exp(q \cdot k_i/\tau)},$$

where k_+ represents the target feature from the alternate view of the same image; k_i denotes a target feature within the key queue; τ is the temperature parameter; and K indicates the size of the key queue, which defaults to 4096.

A linear classifier on ImageNet-1K applied on MoBy with a Swin-T backbone achieves a 75.0% top-1 accuracy whereas DINO with a ViT-S backbone (trained on the same number of epochs) achieves a slightly better result of 75.9%. Swin-T and ViT-S have similar complexity. Neither of these two SSL frameworks surpass the supervised performance on ViT or Swin-T which accounts to 79.8% and 81.3%, respectively. Furthermore, the ability of MoBy to be transferred to object detection and semantic segmentation tasks was also analyzed. The results show that supervised Swin-T transformers perform slightly better than MoBy on both tasks Xie et al. (2021).

To determine how well a self-supervised Swin-T transformer performs on the downstream task of population estimation, we use the pretrained MoBy (Swin-T) model made available by Xie et al. (2021). The particular model has been trained for 300 epochs on ImageNet-1K, with the initial 5 epochs serving as a linear warm-up stage. A batch size of 512 is adopted. Moreover, the AdamW optimizer is employed

with a learning rate of 0.001 and a fixed weight decay of 0.05. An asymmetric drop path regularization has also been proven effective by the authors and thus adopted in pretraining; it is set to 0.1 for the online network and 0.0 for the target network. The starting value for the momentum term is 0.99 and it is gradually increased to 1 during training. Finally, as mentioned above, the size of the key queue is 4096.

3.2.6 MAE - Masked Autoencoder

The Masked Autoencoder (MAE) (He et al., 2022) is a generative SSL approach that works with a ViT backbone. MAE learns visual representations through masking random patches from the input image and then reconstructing the missing patches at pixel-level. First, the image is divided into non-overlapping patches and a subset of these patches is randomly sampled without replacement to be masked. The architecture consists of an asymmetric encoder-decoder design. The encoder is a ViT that is only applied on the visible, unmasked subset of patches. The decoder is another lightweight transformer which takes as input both the encoded visible patches and the mask tokens. Each mask token is a learned vector that signifies the presence of a masked patch that needs to be predicted. Positional embeddings are added to all patches, so that mask tokens will have information regarding their location in the image.

The reconstruction target is set to be the normalized pixel values of each masked patch. The output of the decoder is an array of pixel values representing a patch. A mean squared error (MSE) loss is then computed between the masked patches of the original (ground truth) and reconstructed images in the pixel-space.

MAE has been shown to perform well when a high percentage of the patches is masked, i.e. under a high masking ratio ($\sim$75%). Due to the fact that the encoder only processes the unmasked patches which makes up a small portion of the image, and that the decoder is lightweight, the pretraining time and memory consumption are largely reduced. This makes MAE easy to scale to larger models that generalize well. He et al. (2022) find that a ViT-Huge model pretrained and finetuned with MAE on ImageNet-1K outperforms all previous results by achieving an accuracy of 87.8%. The representation learning capabilities of MAE have also been studied through a series of transfer learning experiments. MAE pretraining with a ViT-B backbone shows to outperform supervised pretraining on both object detection and semantic segmentation downstream tasks.

We evaluate the performance of a number of different pretrained MAE models on our downstream task of population estimation. Next, we describe the implementation details of these models.

3.2.6.1 MAE (ViT-B/16) pretrained on ImageNet

We obtain a MAE model with a ViT-B/16 backbone pretrained on ImageNet by (He et al., 2022). The pre-training setting uses the AdamW optimizer with a base learning rate of 0.00015 and a weight decay of 0.05. The optimizer's momentum parameters are set to $\beta_1 = 0.9$ and $\beta_2 = 0.95$. This model has been trained for 800 epochs with a batch size of 4096. The first 40 epochs are used as warm-up where the learning rate scaler is defined LearningRate = 0.00015 $\times$ BatchSize/256. Then, the learning rate schedule

follows a cosine decay pattern. For data augmentation, RandomResizedCrop and a mask ratio of 0.75 is applied.

3.2.6.2 MAE (ViT-B/16) pretrained on Potsdam

Muhtar et al. (2023) make available a MAE (ViT-B/16) model trained on Potsdam [3.1.6]. They trained with a batch size of 64 for 400 epochs. All other training settings are identical to 3.2.6.1.

3.2.6.3 MAE (ViT-B/16) pretrained on MillionAID

The authors of Wang et al. (2022b) released a ViT-B/16 MAE pretrained network on the RS dataset MillionAID [3.1.5]. Due to the significant distribution differences between natural images and remote sensing images (RSIs), the authors explore the optimal mask ratio in MAE for RSIs. The result was the same as above; they observed that a mask ratio of 0.75 performs best on RSIs too. Finally, the network was trained for 1600 epochs with all other hyperparameters set to the MAE default detailed in Section 3.2.6.1.

3.2.7 DINO-v2

The research from (Oquab et al., 2023) focuses on building a foundation model for computer vision which would learn general-purpose features. If such a model could be pretrained on a large curated dataset then it would benefit numerous downstream tasks. The authors introduce a discriminative ViT self-supervised model, DINO-v2, and a large-scale curated dataset, LVD-142M (Section 3.1.9). Their goal was to create a model that learns features both at the image-level and patch-level which could stabilize well when scaling in both model and data sizes.

DINO-v2 is a combination of DINO (Section 3.2.4) for image-level representations, and iBOT (Zhou et al., 2021) for patch-level representations and some new components described below. Details for the DINO framework can be found on the dedicated section above. Here, we define the DINO loss term as follows:

$$L_{\text{DINO}} = -\sum P_t \log P_s,$$

where P_t and P_s are the output probability distributions of the teacher and student networks, respectively. For the patch-level objective, a random mask is applied to input patches given to the student network, but not the teacher network. The student iBOT head is then applied to the student mask tokens, and similarly the teacher iBOT head is applied to the visible teacher patch tokens that correspond to the ones that are masked in the student network. A softmax normalization and centering are then applied, obtaining the iBOT loss term:

$$L_{\text{iBOT}} = -\sum_i P_{ti} \log P_{si},$$

where i are the patch indices of masked tokens. Similar to DINO, the exponential moving average of past iterations is used to learn the parameters of the student network and build the teacher head.

A separate learnable MLP projection head is applied to each of the DINO and iBOT output tokens and the loss is calculated atop. The authors call this feature of the model 'untying heads'. Moreover, the teacher softmax-centering step of DINO and iBOT is replaced by the Sinkhorn-Knopp (SK) batch normalization (Caron et al., 2020). To spread features uniformly within a batch, the KoLeo regulaizer (Sablayrolles et al., 2018) is adopted. Given a set of n vectors,

$$L_{\text{koleo}} = -\frac{1}{n}\sum_{i=1}^{n}\log(d_{n,i}),$$

where $d_{n,i} = \min_{j\neq i}\|x_i - x_j\|$ is the minimum distance between x_i and any other point within the batch.

Another key detail of DINO-v2 is that the resolution of the images is increased to 518x518 from 224x224 for a short period at the end of the pretraining process. Increasing the image resolution is crucial for pixel-level downstream tasks like segmentation or detection, as small objects can vanish at lower resolutions. However, training with high-resolution images is both time-consuming and memory-intensive.

Finally, for small architectures, the authors distill larger models instead of training them from scratch in a similar process to the teacher-student knowledge distillation method described above. However, some modifications were made: a larger, frozen model is utilized as the teacher, an exponential moving average (EMA) of the student model is maintained and used as the final model, masking is removed, and the iBOT loss is applied on the two global crops. Their ablation studies have shown that this approach outperforms training from scratch, even when using a ViT-L model.

Oquab et al. (2023) evaluate the performance of DINO-v2 (pretrained on LVD-142M [3.1.9]) on several image understanding tasks. The results show that the pretrained DINO-v2 model with a ViT-G/14 outperforms the SSL methods of MAE, DINO and iBOT on both linear evaluation on ImageNet-1K and downstream tasks, such as image classification, instance-level recognition, semantic segmentation and depth estimation. The comparison was made across architectures and pretraining data. More specifically, DINO-v2 (ViT-G/14) achieved the new state-of-the-art performance on linear evaluation on ImageNet-1K, surpassing iBOT (ViT-L/16 reained on ImageNet-22K) by 4.2%. A DINO-v2 model with a smaller architecture of ViT-B/14 also achieves better performance than the old state-of-the-art by 2.2%.

We use the DINO-v2 ViT-B/14 (Oquab et al., 2023) model trained (using knowledge distillation from a more complex model) on LVD-142M as a pretraining model in our study. The model is trained for 625,000 iterations using the AdamW optimizer, with an initial LayerScale value of 1×10^{-5}. The weight decay follows a cosine schedule ranging from 0.04 to 0.2, and the learning rate is warmed up over the first 100,000 iterations. The teacher momentum also follows a cosine schedule, starting from 0.994 and increasing to 1. For high-resolution adaptation, the model is initialized with pretrained weights and trained for 10,000 iterations, using the same procedure and schedules as the original pretraining, but compressed to fit the shorter duration and with a reduced base learning rate.

3.2.8 The Original Vision (TOV) model

The Original Vision (TOV) model for remote sensing image understanding (RSIU) developed by Tao et al. (2023b) is a self-supervised vision model that uses both natural and remote sensing images (RSIs) for training. The authors of TOV argue that vision models for RSIU would learn better if they were taught in a 'human-like' way, i.e., to first learn general knowledge from natural images and then acquire the domain-relevant specialized knowledge through remote sensing images. This method is specialized to RS tasks but the authors choose not to train on RS images from scratch, as they suspect that adding the level of general knowledge learning would help the model to generalize better since natural images have higher resolution and richer texture details than RSIs.

Thus, TOV training consists of two stages: (1) the general knowledge stage and (2) the specialized domain-specific stage. Contrastive SSL is performed on Stage 1 and the parameters learned are used to initialize a contrastive model in Stage 2. To avoid the problem of catastrophic forgetting (Goodfellow et al., 2013), TOV applies a memory retention strategy which involves fixing the weights of the shallow and middle layers of the network learned in Stage 1. These layers typically capture low-level visual features such as edges, textures, and basic shapes, and by fixing these weights, the network retains the knowledge of these low-level features (Yosinski et al., 2014). The objective function for this optimization is represented as:

$$\min_{\{w|w\notin W_b \& w\in W_b'\}} L$$

where W_b represents the weights of the fixed layers, W_b' represents the weights being optimized and L is the SSL contrastive loss function (Tao et al., 2023b).

The data acquisition method for the TOV model was designed to gather a large-scale, diverse dataset that includes both natural images and remote sensing images. These datasets have been given the names TOV-NI and TOV-RS, respectively. Details regarding the nature of these datasets have been provided in Section 3.1.7.

Tao et al. (2023b) provide performance results for TOV against supervised ImageNet pretraining and other popular self-supervised learning methods such as SimCLR (Chen et al., 2020a), MoCo-v2 (Chen et al., 2020b), SSL4EO (Wang et al., 2023) and SeCo (Manas et al., 2021). TOV was pretrained first on TOV-NI and then TOV-RS, following the 'human-like pretraining' as mentioned above. The SimCLR and MoCo models were only pretrained on TOV-RS. Finally, SeCo and SSL4EO were pretrained on their respective datasets. A ResNet50 backbone and the same training hyperparameters have been used for all models. The authors evaluate the generalization capabilities of these models on three remote sensing image understanding tasks: scene classification, semantic segmentation and object detection on several specialized datasets for each task. The TOV method consistently outperforms ImageNet supervised pretraining and the SSL models in the majority of the 12 benchmark RS datasets used for evaluation across the three downstream tasks.

Tao et al. (2023b) provide a publicly available trained TOV model with a ResNet50 backbone which is used in this work as the pretrained model. The training was conducted using the Adam optimizer with a batch size of 1024. The learning rate was initially

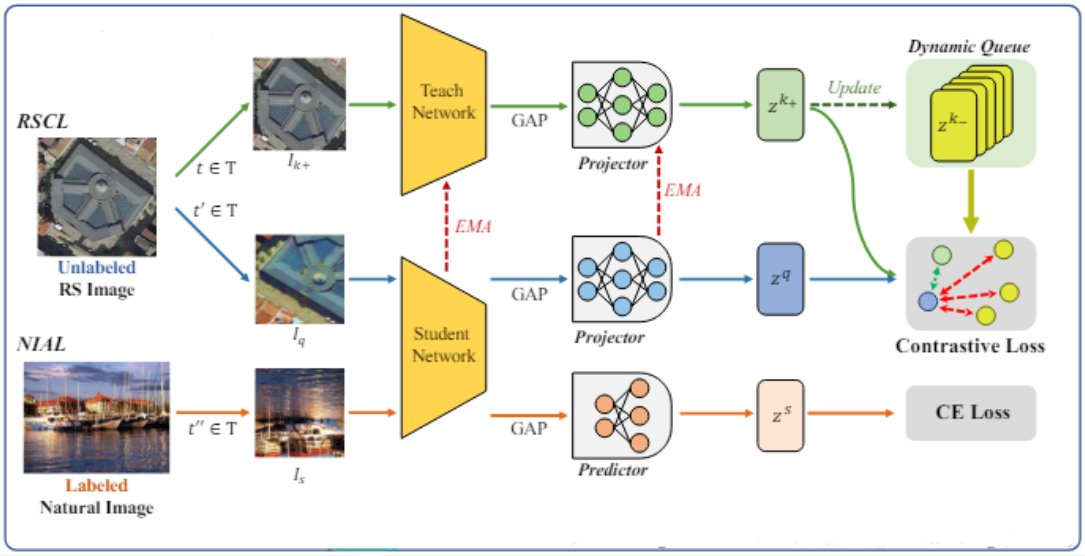

Figure 3.4: Pretraining process of Generic Knowledge Boosted Remote Sensing Pretraining (GeRSP). Taken from Huang et al. (2024).

set at 0.75 and gradually decreased following a cosine schedule over the course of 800 epochs.

3.2.9 GeRSP – Generic Knowledge Boosted Remote Sensing Pretraining

GeRSP (Huang et al., 2024), or Generic Knowledge Boosted Remote Sensing Pretraining, is similar to TOV, since it also makes use of both remote sensing images and natural images. A core difference between the two however is that GeRSP uses supervised learning when training on natural images and self-supervised learning when training on remote sensing images, whereas TOV uses solely SSL.

More precisely, GeRSP is made up of two learning processes: natural image auxiliary learning (NIAL) on labeled natural images and remote sensing contrastive learning (RSCL) on unlabeled remote sensing (RS) images. Inspired from the success of SimCLR [3.2.1], the GeRSP framework adopts a strong augmentation strategy for both RSCL and NIAL to achieve feature invariance.

As shown in Figure 3.4, GeRSP makes use of a teacher-student network. RSCL trains both the teacher and student network at the same time, while NIAL only trains the student network. During training, the teacher network is updated with an exponential moving average of the student network parameters, whereas the student network learns from the contrastive loss calculated between the augmented views of the same image (positive pairs) and different images (negative pairs). The student network, after being trained on both remote sensing and natural images, serves as the pretrained model for downstream tasks. In each iteration of GeRSP, an equal number of natural and remote sensing images are used from their respective datasets.

The backbone network used by GeRSP is ResNet50. The extracted features are ag-

gregated using global average pooling (GAP). The RSCL pipeline then employs a non-linear projector, whereas the NIAL pipeline employs a predictor. The projector consists of two fully-connected layers with ReLU activation, with a hidden dimension of 2048 and an output dimension of 128. On the other hand, the predictor is a single fully-connected layer that maps features to logits for classification. This distinction between the projector and predictor helps in addressing the potential conflicts between self-supervised and supervised learning tasks within the same framework. By using a dedicated predictor for supervised learning, GeRSP can maintain the task-specific characteristics needed for accurate classification while still utilizing the generalized features learned through contrastive learning.

The cross-entropy loss is used to optimize NIAL, whereas, RSCL utilizes the InfoNCE loss as the contrastive loss function. Inspired from MoCo (He et al., 2020) (see Section 3.2.2), GeRSP's teacher network uses momentum update on the network parameters to ensure a stable learning process. Let W_t and W_s represent the parameters of the teacher and student networks respectively. The update rule for W_t is:

$$W_t \leftarrow mW_t + (1-m)W_s$$

where the momentum coefficient m is set to 0.996. The dynamic queue is implemented as a First-In-First-Out (FIFO) queue and stores features generated by the teacher network. After each optimization iteration, both the parameters of the dynamic queue and the teacher network are updated.

The authors of GeRSP have evaluated its performance against ImageNet supervised pretraining, and other self-supervised pretrained models, such as TOV [3.2.8], MoCo-v2 [3.2.2] and MoCo pretrained on ImageNet, MillionAID or ImageNet + MillionAID (similar to GeRSP's training procedure). GeRSP performs better than all the forementioned models on the downstream tasks of scene classification and semantic segmentation. Finally, it seems that GeRSP generalizes better than pure contrastive methods and TOV as it shows transferability across various datasets and tasks.

Huang et al. (2024) have made publicly available a GeRSP model trained on the unlabeled RS image dataset Million-AID (Long et al., 2021) [3.1.5] and the labeled natural image dataset ImageNet [3.1.8]. This model was trained with the stochastic gradient descent (SGD) optimizer for 200 epochs. The initial values of the learning rate was 0.05, 0.90 for weight decay, and 0.00005 for momentum. A cosine annealing scheduler was used to optimize the learning rate. This model has been used for pretraining in this work.

3.2.10 CMID - Contrastive Mask Image Distillation

Muhtar et al. (2023) have created a knowledge distillation model which combines contrastive learning with masked image modelling (MIM, a generative approach similar to MAE [3.2.6]) called CMID. This framework is similar to that of DINO-v2 [3.2.7], but CMID was designed specifically for remote sensing image understanding. They argue that neither solely contrastive nor solely MIM approaches are capable to transfer well to remote sensing understanding tasks since contrastive SSL methods are limited to

learning inter-image, or global, representations, whereas MIM approaches are limited to learning intra-image, or local, representations.

CMID consists of three branches: the MIM branch, the global branch, and the local branch. The MIM branch employs the MIM method to learn local spatial context. The global branch uses contrastive learning, specifically the MoCo framework [3.2.2], to learn global semantic information. The local branch, on the other hand, focuses on recovering object-level information lost in the MIM branch through knowledge distillation. As seen in Figure 3.5, through this teacher-student architecture, the entire network interacts and stabilizes between the different learning signals of each branch.

Given an image x, CMID first creates two versions of x: a masked image and an augmented image, using either a mask augmentation or random data augmentations. These are then input into the student and teacher networks, respectively. Both networks share the same architecture (either CNN or ViT). The student's parameters are updated to be an exponential moving average of the teacher's parameters. The student maps the masked image to a latent embedding, while the teacher encodes the augmented image to preserve the semantics of image x and guide the student with contrastive learning.

3.2.10.1 Masked Image Modeling Branch

The MIM branch follows an adaptation of the SimMIM (Xie et al., 2022) framework. The main differences between SimMIM and MAE are: (i) SimMIM includes the masked patches in the encoder too, and (ii) SimMIM uses a 'one-layer prediction head' as a decoder, unlike MAE which uses a transformer. Even though SimMIM is much simpler than MAE, it still shows competitive performance on many downstream tasks.

Remote sensing (RS) images are known for having multiple objects that are usually densely distributed. Losing these objects through the masking operation might lead to incomplete semantic meaning, substantially increasing the semantic discrepancy between masked and augmented images and making this a very challenging image reconstruction task. Therefore the authors have decided that instead of zero-initialized mask tokens, they are to use the mean spectral value to fill the learnable mask tokens with. The resulting image, x_{mask} is expressed as follows:

$$x_{\text{mask}} = x_m^P + \text{MASK} \odot M$$

where x_m^P represents the set of image patches once the masked patches are initialized with the mean spectral value. This serves as input to the SimMIM encoder which generates a latent representation of the masked image. Then, the one-layer MLP decoder uses this embedding to produce a predicted reconstruction image x'.

The model parameters are then updated through an l_1 loss L_{spat} applied to the masked pixels

$$L_{\text{spat}} = \frac{1}{\Omega(x_m)} \|x_m - x'_m\|_1,$$

where x_m and x'_m represent the sets of the original and reconstructed pixel values of the masked patches, respectively, and $\Omega(\cdot)$ denotes the number of elements in a set. Finally, to enforce consistency among the original and the reconstructed image, the authors

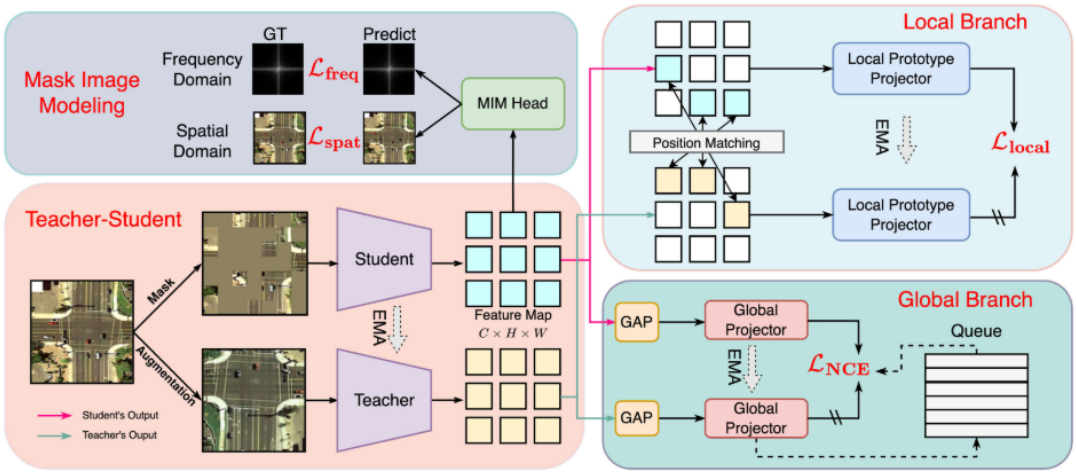

Figure 3.5: The pipeline of CMID. Taken from Muhtar et al. (2023).

decide to incorporate the focal frequency loss (FFL) (Jiang et al., 2021) L_{freq} in the frequency domain. FFL has shown to be effective in learning high-level semantics. The focal frequency loss is defined as follows:

$$L_{\text{freq}} = \frac{1}{N} \sum_{c=1}^{N} \text{FFL}(x_c, x'_c),$$

where N denotes the number of input channels for image x, and c refers to the specific channel of the image.

3.2.10.2 Global Branch

While the MIM branch focuses on capturing fine-grained visual representations of the images, the global branch aims to recover the global semantic content of the masked image by aligning the student and teacher representations using a visual dictionary queue. The global branch adopts the MoCo contrastive learning method, where the outputs of the student and the teacher networks serve as the query and the key respectively. The query and the key both go through a round of global average pooling (GAP) and a global projection which embeds them into a global representation space. Then the infoNCE loss [3.3] is applied to create better representations by discriminating them against other images. The queue is made up of the teacher's projected global representations and is updated at each iteration.

3.2.10.3 Local Branch

The local branch's aim is to further combat semantic incompleteness by aligning together the local semantics of the student and teacher. N position-matched pairs are selected from the feature vectors of the student's and teacher's output feature maps using their absolute positions in the original input image x, represented as $\{(x_i, \hat{x}_i)\}_{i=1}^{N}$. The matched pairs are then projected onto a different feature space and mapped to a set of learnable prototypes to determine p_i and q_i; the similarity distributions between

x_i and $\hat{x}_i$ with respect to the prototypes. Cross-entropy loss is applied to minimize the difference between p_i and q_i:

$$L_{\text{local}} = \frac{1}{N} \sum_{i=1}^{N} -p_i \log q_i$$

Therefore, the total loss of CMID is defined as

$$L = \lambda_1 (L_{\text{spat}} + L_{\text{freq}}) + \lambda_2 L_{\text{NCE}} + \lambda_3 L_{\text{local}}$$

where λ_1, λ_2, and λ_3 serve as weights to balance the three branches.

The authors have evaluated how well CMID pretraining on Potsdam [3.1.6] performs in several downstream tasks. CMID, with either a CNN or transformer backbone, performs better than other self-supervised pretrained models, such as BYOL [3.2.3], Barlow Twins (Zbontar et al., 2021), MoCo-v2 [3.2.2], MAE [3.2.6] and SimMIM (Xie et al., 2022), in both semantic segmentation (evaluated on Potsdam) and object detection tasks (evaluated on DOTA (Xia et al., 2018)). Moreover, CMID outperforms supervised ImageNet pretraining and state-of-the-art self-supervised models in scene classification tasks. This highlights the ability of CMID to scale and generalize better in the remote sensing domain.

We obtain four self-supervised pretrained CMID models from Muhtar et al. (2023) pretrained on Potsdam [3.1.6] and MillionAID [3.1.5] separately. A ResNet-50 or Swin-B architectures are used as the encoder for the student and teacher networks.

3.2.10.4 CMID models pretrained on Potsdam

Both the CMID-ResNet50 and CMID-Swin-B models are trained on the Potsdam dataset from scratch for 400 epochs. A batch size of 64 and an Adan optimizer with a learning rate and a weight decay set to 0.003125 and 0.02, respectively, was employed.

3.2.10.5 CMID models pretrained on MillionAID

A CMID-ResNet50 and a CMID-Swin-B model were trained on MillionAID for 200 epochs, with a batch size of 512 for ResNet50 and 256 for Swin-B. An Adan optimizer was used, with a learning rate of 0.0088 and 0.002 for ResNet50 and Swin-B, respectively. The weight decay was set to 0.02. Moreover, a cosine learning rate scheduler was employed, and the first 15 epochs are used as a warm-up phase.

3.3 Population Estimation with SCIPE

In this section, we introduce the datasets used for the downstream task of population estimation, discuss the process we follow in estimating population, and propose the methods used to evaluate the performance of the models.

3.3.1 Microcensus Survey Data and Satellite Imagery

Microcensus We utilize a UNICEF-funded microcensus from 2019 among two districts in Mozambique: Magude (MGD) and Boane (BOA). The survey was conducted at a household-level, meaning that the population of each individual building was surveyed. The geo-location of each household is also made available. The household survey data was aggregated into a 100-meter grid, creating 474 labeled grid tiles for population counts.

Satellite Imagery In addition to the microcensus data, we make use of very-high-resolution (0.5 m) satellite imagery covering 7773 km^2 across the two districts (BOA and MGD). This data is part of the Vivid 2.0 data product and was acquired from Maxar. The satellite images in Vivid 2.0 only consist of the three RGB bands (red, green, and blue). This product's images are updated annually but images that have high cloud coverage are getting replaced by an older corresponding image that has a lower cloud coverage. Therefore, images can be from a set of different time periods. Our data is a mosaic of images mostly from 2018 and 2019.

Due to the temporal misalignment between the imagery and microcensus many tiles contained either unsurveyed buildings or buildings absent in the imagery. Such mismatched tiles are considered 'outliers'. Thus, Neal et al. (2022) have manually examined each grid tile, comparing the GPS locations of surveyed buildings with those visible in the imagery. Once outlier tiles have been excluded, the tiles to which we have ground truth labels sum up to 199.

Another issue with our labeled data is that we don't explicitly have tiles with zero population. This is because, naturally, the microcensus was only conducted in populated areas. If we are not to include uninhabited tiles into our dataset, we would expect that our model will systematically overestimate as it would fail to predict zero population. To address this, Neal et al. (2022) have manually identified 75 random tiles (25 from MGD, 50 from BOA) with zero population based on the HSRL population map. The SCIPE model makes use of these zero-population tiles during training. However, we have observed that self-supervised models perform better without the use of these tiles. We report the performance of our models with and without training on zero-population tiles in Section A and Section 4, respectively.

3.3.2 Self-supervised SCIPE

We used pretrained CNN and transformer architectures, described in Section 3.2, trained on either remote sensing datasets, natural image datasets, or both.

We first log-transform our targets and resize our tiles accordingly. Then we feed the representations extracted from each pretrained model to a separate Random Forest regressor to train our prediction model. The model's hyperparameters were selected through a grid search, exploring the following ranges: the number of estimators $\{100, 200, ..., 500\}$, the minimum number of samples required to split an internal node $\{2, 5\}$, and the minimum number of samples required to be at a leaf node $\{1, 2\}$.

3.3.2.1 Finetuning

We finetune the pretrained models using our microcensus grid tiles by attaching a linear regression head and minimizing the L2 loss between the observed and predicted population. We should note here that only the pretrained models are trained in a self-supervised fashion; finetuning is supervised. The 199 labeled grid tiles were randomly split into training and validation sets (80-20%). Due to the limited number of tiles in the dataset, we applied random dihedral transformations (i.e., reflections and rotations) to augment the training set, avoiding any transformations that could compromise the validity of the population count, such as cropping that might remove buildings. We used the Adam optimizer (with default settings) to minimize the loss function with a batch size of 32.

During training, we first froze the network and trained only the regression head for 25 epochs with a learning rate of 2×10^{-3}. Next, we trained the entire network using a discriminative learning rate approach, where the learning rate was higher at the top of the network and reduced in the earlier layers. This method avoided large changes to the earlier layers, which typically extract more general features, and focused training on the domain-specific later layers. The base learning rate at the top of the network was 1×10^{-3}, decreasing to a minimum of 1×10^{-5} in the earlier stages. Early stopping was used to halt training when the validation loss had no improvement for 2 or more epochs to avoid overfitting.

3.3.3 Evaluation and Cross-validation

We evaluate the different methods using several metrics. In the following metrics, let y_i represent the actual value of the i-th observation, $\hat{y}_i$ represent the predicted value of the i-th observation, and $\bar{y}$ represent the mean of the actual values.

R^2 Score

The R^2 score measures the proportion of variance in the training set that is captured by the model's predictions. It serves as a "goodness of fit" indicator, with higher values indicating better performance.

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y}_i)^2}$$

Median Absolute Percentage Error (MeAPE)

MeAPE calculates the percentage difference between each predicted and actual value, then returns the median of these values. This metric is robust against outliers, and lower values indicate better performance.

$$\text{MeAPE} = \text{median}\left(100 \times \frac{|y_i - \hat{y}_i|}{y_i}\right)$$

Adjusted Median Absolute Percentage Error (AdjMAPE)

AdjMAPE is a modified version of MeAPE that adjusts the calculation (using k; a small constant) to handle cases where the actual values are zero or near zero, preventing infinite or undefined percentage errors. This adjustment ensures meaningful and finite errors.

$$\text{AdjMAPE} = \text{median}\left(\frac{|y_i - \hat{y}_i|}{y_i + k}\right)$$

Median Absolute Error (MeAE)

MeAE represents the median absolute difference between all predicted and actual values. Like MeAPE, it is robust against outliers, with lower values indicating better accuracy.

$$\text{MeAE} = \text{median}(|y_i - \hat{y}_i|)$$

Aggregate Percentage Error (AggPE)

Similar to MeAPE, AggPE calculates the percentage error of the model at the aggregate level, rather than on a per-instance basis. Lower values are preferred.

$$\text{AggPE} = \frac{|\sum_i y_i - \sum_i \hat{y}_i|}{\sum_i y_i}$$

With the exception of AdjMAPE, all the above metrics have been used by Neal et al. (2022) to evaluate the performance of SCIPE. Therefore, we choose to use the same ones so that we can directly compare our, self-supervised SCIPE, performance to the original SCIPE. We chose to include the AdjMAPE metric too since it provides a measure that is particularly useful when the dataset includes zero or near-zero actual values.

Cross-validation

In addition, we use cross-validation to evaluate our different population estimation methods. For each region, we divided the data into four spatial subsets and created validation folds by combining these subsets across the two regions. We reported the evaluation metrics based on pooled predictions from the four validation folds, covering the entire microcensus. To avoid data leakage, we trained a separate network for each fold, resulting in four distinct networks.

Chapter 4

Experiments and Results

In this chapter, we compare the performance of several self-supervised learning methods using cross-validation. Extensive experimentation was conducted to investigate the influence of different pretraining paradigms on census-independent population estimation. We explore the effect of various factors, including different learning signals, such as supervised versus self-supervised methods and contrastive versus masked image modeling (generative) approaches. Additionally, we examine dataset characteristics, focusing on the domain of images (natural images versus remote sensing images), sample size, and spatial resolution. Lastly, we consider backbone network characteristics, comparing architectures like ResNet-50, Swin, and ViT, as well as network capacity.

We take the SCIPE (Neal et al., 2022) model to be our baseline. SCIPE is a finetuned Barlow-Twins model pretrained in a supervised manner on ImageNet, with a ResNet-50 backbone. We use publicly available pretrained models for all our experiments, to which implementation details and citations are found in Section 3.2. Pretraining our own models would need resources and time that were out of budget for this project. Thus, there are some inconsistencies in our experiment design: (1) we do not assess a supervised ViT model pretrained on remote sensing images (RSI), (2) we only evaluate supervised RSI pretraining on one dataset - Million-AID - and, (3) we evaluate a 'base' Swin transformer (Swin-B) for SSL RSI performance instead of a 'tiny' Swin transformer (Swin-T) like in other categories.

Tables 4.1, 4.2, and 4.3 show the cross-validation results for population estimation using Random Forest regression on representations extracted by the pretrained models with non-outlier and non-zero-population tiles only [3.3.1]. Tables A.1, A.2, and A.3 show the cross-validation results on representations extracted by the pretrained models with non-outlier and zero-population tiles. The results that include zero-population tiles show an overall poorer performance, especially on the aggregate percentage error (AggPE), thus any further discussion on performance refers to the results on non-outlier and non-zero population tiles.

The first observation we make is that while finetuning pretrained models with micro-census improves the performance for most models, all finetuned models with a Swin backbone show decreased performance. Moreover, all the finetuned ResNet-50 mod-

Model	Dataset	Backbone	# Params	R^2	MeAPE	AdjMAPE	MeAE	AggPE
Supervised (NI)								
SCIPE (BarlowTwins FT)	ImageNet	ResNet-50	24M	0.39	44.9%	–	3.91	01.1%
Swin-T	ImageNet	–	28M	0.14	54.3%	0.25	4.38	14.5%
Swin-T FT	ImageNet	–	28M	0.26	45.6%	0.22	*3.69*	15.4%
ViT-B/16	ImageNet	–	86M	0.15	53.4%	0.25	4.49	16.7%
ViT-B/16 FT	ImageNet	–	86M	***0.44***	***42.2%***	**0.19**	***3.34***	***00.7%***
Supervised (RSI)								
ResNet-50	Million-AID	–	24M	**0.04**	56.3%	0.25	**4.25**	15.2%
ResNet-50 FT	Million-AID	–	24M	-0.13	**55.1%**	**0.24**	4.43	**03.4%**
Swin-T	Million-AID	–	28M	0.01	58.5%	0.29	4.75	16.2%
Swin-T FT	Million-AID	–	28M	-0.04	60.7%	0.29	4.90	16.3%

Table 4.1: Performance of the supervised pretrained models on natural images (NI) or remote sensing images (RSI). FT stands for finetuned model. The best results of each category are shown in bold. The best results overall are shown underlined. Results that outperform the baseline, SCIPE, are shown in italics.

els pretrained with SSL on natural images (Table 4.2) and the MoCo-v2 (FT) model pretrained on Potsdam, performed poorer in all evaluation metrics except AggPE.

Furthermore, we observe that a supervised ViT-B/16 model pretrained on ImageNet outperforms SCIPE in all evaluation metrics, while performing the best MeAPE overall (Table 4.1). Self-supervised (SS) pretraining on remote sensing images also improves SCIPE's performance and achieves the best results overall in all evaluation metrics except MeAPE. The three SS SimCLR models pretrained on RSIs provide the best per-tile population estimation, outperforming SCIPE in all evaluation metrics except aggregate percentage error (AggPE). SimCLR UC Merced and SimCLR SIRI-WHU have the largest R^2 metric overall ($R^2 = 0.53$), whereas SimCLR MLRSNet performs the best on the AdjMAPE and MeAE evaluation metrics (AdjMAPE = 0.17, MeAE = 3.16). Conversely, with consecutive self-supervised pretraining on NI and RSI, TOV shows the best AggPE overall, 0.5% lower than SCIPE.

4.1 Effect of Learning Signal

Our experiment results agree with Tao et al. (2023a), showing that contrstive learning methods learn better representations than masked image modeling approaches. We notice from tables 4.2 and 4.3 that the MIM approach, MAE, performs much poorer than contrastive approaches, except in the case where the contrastive models have a Swin transformer as their backbone network.

Methods that combine contrastive and MIM learning signals, such as DINO-v2 and CMID (with a ResNet-50 backbone), perform well, too, irrespective of the domain of the pretraining dataset. DINO-v2, which is pretrained on a very-large-scale NI dataset, is the best-performing SS NI model, outperforming SCIPE in the MeAE and MeAPE

Model	Dataset	Backbone	# Params	R^2	MeAPE	AdjMAPE	MeAE	AggPE
Self-Supervised (NI)								
SimCLR	ImageNet	ResNet-50	24M	0.12	52.6%	0.26	4.26	14.6%
SimCLR FT	ImageNet	ResNet-50	24M	-0.15	60.4%	0.29	4.95	06.0%
MoCo-v2	ImageNet	ResNet-50	24M	0.15	54.3%	0.24	4.52	14.2%
MoCo-v2 FT	ImageNet	ResNet-50	24M	-0.07	58.6%	0.28	4.70	05.8%
MAE	ImageNet	ViT-B/16	86M	-0.01	59.0%	0.26	4.67	17.3%
MAE FT	ImageNet	ViT-B/16	86M	0.04	54.7%	0.26	4.37	09.7%
DINO	ImageNet	ResNet-50	24M	0.04	56.6%	0.28	4.60	15.4%
DINO FT	ImageNet	ResNet-50	24M	-0.04	58.9%	0.28	4.80	07.1%
DINO	ImageNet	ViT-B/16	85M	0.04	57.9%	0.27	4.51	15.1%
DINO FT	ImageNet	ViT-B/16	85M	0.16	51.8%	0.25	4.60	07.7%
DINO-v2	LVD-142M	ViT-B/14	86M	0.13	53.3%	0.25	4.23	17.0%
DINO-v2 FT	LVD-142M	ViT-B/14	86M	**0.38**	***44.0%***	**0.20**	***3.54***	**04.3%**
MoBy	ImageNet	Swin-T	28M	0.01	60.7%	0.29	4.78	16.8%
MoBy FT	ImageNet	Swin-T	28M	0.02	57.7%	0.27	4.54	17.5%

Table 4.2: Performance of the self-supervised pretrained models on natural images (NI). FT stands for finetuned model. The best results of each category are shown in bold. The best results overall are shown underlined. Results that outperform the baseline, SCIPE, are shown in italics.

evaluation metrics (MeAPE = 44.0%; MeAE = 3.54). Whilst, CMID pretrained on Million-AID has the second best R^2 overall (R^2 = 0.52), and CMID pretrained on Potsdam surpass SCIPE evaluations in both R^2 and MeAE (R^2 = 0.46; MeAE = 3.90). All CL+MIM models (besides CMID with a Swin backbone) surpass MAE - a solely MIM approach - irrespective of data domain.

Moreover, we observe that when pretrained on RSIs, self-supervised models learn more transferable representations than supervised models since many of the finetuned SS RSI models outperform SCIPE in MeAE and R^2 metrics, whereas supervised RSI models are far behind. On the other hand, supervised NI pretraining performs better than SS NI pretraining. The ViT-B/16 model pretrained on ImageNet outperforms SCIPE and all SS NI pretraining paradigms in all evaluation metrics.

An unexpected observation is that supervised pretraining on RSIs performs the poorest across all categories. Both supervised RSI models are pretrained on Million-AID, so we could also blame the dataset for the poor performance. However, SS RSI models pretrained on Million-AID perform well. This leads us to think that, as Tao et al. (2023b) discussed, the inherent information contained in the vast amounts of remote sensing data is theoretically more abundant and essential than the semantic information derived from human-labeled samples.

4.2 Effect of Data Domain, Size and Resolution

Please note: Table 3.1 summarises the characteristics of each dataset.

As already mentioned above, remote sensing (RS) datasets, pretrained in a self-supervised (SS) manner, show improved performance from models of any pretraining form on natural images (NIs). However, one exception exists; ViT-B/16 pretrained in a supervised manner on ImageNet performs better on the aggregate percentage error (AggPE) metric than any model pretrained on solely RS images. TOV, which is SS pretrained on NIs first and on RSIs next - in a consecutive manner (explained in 3.2.8) - outperforms supervised ImageNet pretraining with ViT-B/16 by 0.1%, scoring the best AggPE across all models (AggPE = 0.6%). This shows the necessity of shallow ImageNet pretraining in acquiring low- and mid-level representations, before pretraining on remote sensing images as argued by Tao et al. (2023b). GeRSP, however, which is also pretrained on both NIs and RSIs - but in the more sophisticated manner of knowledge distillation - does not show the improved AggPE performance that TOV does. This could be due to the difference in dataset choice. The spatial resolution of TOV-RS ranges from 1 to 20 m, whereas the spatial resolution of Million-AID ranges from 0.5 to 153 m. The lowest resolutions of the datasets have a much bigger disparity than the highest resolutions do, which leads us to assume that TOV-RS is a dataset of higher resolution. This characteristic of the TOV pretrained model could have contributed to its better performance against GeRSP, as we will discuss later.

Research in the field of natural image classification has proven many times that training with larger and more diverse datasets learns better representations (Wang et al., 2022a). This is also illustrated in Table 4.2, where DINO-v2 which is pretrained on a dataset $111\times$ larger than ImageNet, performs the best within the category. Would that be the case for remote sensing pretraining too? We observe that self-supervised models pretrained on RSIs achieve the best results when the pretraining dataset has a very high resolution, even in cases where the number of samples in the dataset are less than 2,500. As we can see in Table 4.3, SimCLR models pretrained on UC Merced, MLRSNet, and SIRI-WHU outperform all other models in non-aggregate evaluation metrics. As seen in Table 3.1, UC Merced has 2,100 samples with a 0.3 m resolution; SIRI-WHU has 2,400 samples with a 2 m resolution; and MLRSNet has 109,161 samples with 0.1 - 10 m resolution. On the other hand, Million-AID has 1,000,848 samples with its resolution ranging from very high (0.5 m) to very low (153 m). Samples of lower resolution have likely hindered the ability of models pretrained on Million-AID to learn good representations despite the large scale of the dataset.

A strange case arose with SS MoCo-v2 pretrained on SSL4EO-S12 and Potsdam. SSL4EO-S12 has a resolution of 10 m per pixel, however, we are unsure the size of the dataset (as discussed in 3.1.4). Potsdam, on the other hand, has a resolution of of 0.5 m and a size of 21,188 samples. We would expect MoCo-v2 when pretrained on Potsdam to perform better than when pretrained on SSL4EO-S12, but it does not. This could be attributed to the difference in domains between the two datasets. As seen in Table 3.1, Potsdam's domain is Urban Germany, whereas SSL4EO-S12 has worldwide images, including Mozambique, as seen in Figure 3.1.

4.3 Effect of Network Architecture and Capacity

Tables 4.1 and 4.2 show that models pretrained on natural images perform better with a ViT-B backbone (of 86M parameters) than a ResNet-50 backbone (of 24M parameters). This can especially be seen in Table 4.2 where DINO pretrained with a ResNet-50 on ImageNet performs much poorer than DINO pretrained with a ViT-B/16 on the same dataset. The case for SS pretraining on RSI is not the same. CNN models perform much better than transformer models in the SS RSI pretraining paradigm. This can be attributed to the data-hungry nature of ViTs and recent research findings on the relationship between model size and pretraining dataset size (Li et al., 2022). These findings suggest that smaller models trained on smaller datasets can achieve higher learning performance, whereas larger models may not perform as well as the smaller ones in such scenarios. This phenomenon is obvious in our experiments too. Table 4.3 shows the results of two ViT-B/16 MAE models pretrained on Million-AID or Potsdam. Million-AID is almost 46 times larger than Potsdam, and performs better on the non-aggregate evaluation metrics when finetuned.

Morever, as mentioned earlier, the Swin transformer models are the worst performers overall. Further research needs to be done to find the reason behind this.

Model	Dataset	Backbone	# Params	R^2	MeAPE	AdjMAPE	MeAE	AggPE
Self-Supervised (RSI)								
MoCo-v2	SSL4EO-S12	ResNet-50	24M	0.03	56.2%	0.26	4.28	14.0%
MoCo-v2 FT	SSL4EO-S12	ResNet-50	24M	*0.40*	46.6%	0.21	*3.83*	04.0%
MoCo-v2	Potsdam	ResNet-50	24M	0.09	55.6%	0.27	4.59	15.8%
MoCo-v2 FT	Potsdam	ResNet-50	24M	-0.08	56.3%	0.27	4.61	07.7%
BYOL	Potsdam	ResNet-50	24M	0.22	53.6%	0.23	4.18	13.4%
BYOL FT	Potsdam	ResNet-50	24M	*0.43*	47.1%	0.22	*3.87*	06.9%
SimCLR	UC Merced	ResNet-50	24M	0.13	54.3%	0.26	4.33	14.7%
SimCLR FT	UC Mefced	ResNet-50	24M	**0.53**	*43.3%*	0.19	*3.54*	03.1%
SimCLR	MLRSNet	ResNet-50	24M	0.11	54.6%	0.27	4.63	13.0%
SimCLR FT	MLRSNet	ResNet-50	24M	*0.46*	***42.6%***	**0.17**	**3.16**	03.6%
SimCLR	SIRI-WHU	ResNet-50	24M	0.04	57.0%	0.28	4.67	13.7%
SimCLR FT	SIRI-WHU	ResNet-50	24M	**0.53**	***42.6%***	0.19	*3.44*	05.5%
MAE	Million-AID	ViT-B/16	86M	-0.01	59.9%	0.28	4.68	14.8%
MAE FT	Million-AID	ViT-B/16	86M	0.17	54.2%	0.24	3.97	10.7%
MAE	Potsdam	ViT-B/16	86M	0.04	60.3%	0.28	4.78	16.1%
MAE FT	Potsdam	ViT-B/16	86M	0.03	55.1%	0.24	4.45	04.7%
TOV	TOV-NI & TOV-RS	ResNet-50	24M	0.07	55.2%	0.26	4.27	15.6%
TOV FT	TOV-NI & TOV-RS	ResNet-50	24M	*0.48*	46.6%	0.21	*3.66*	**00.6%**
GeRSP	ImageNet & MillionAID	ResNet-50	24M	0.22	52.7%	0.24	4.17	15.9%
GeRSP FT	ImageNet & MillionAID	ResNet-50	24M	*0.47*	45.9%	0.20	*3.60*	07.6%
CMID	Potsdam	ResNet-50	24M	0.27	48.4%	0.23	*3.76*	16.9%
CMID FT	Potsdam	ResNet-50	24M	*0.46*	45.7%	0.22	*3.90*	01.7%
CMID	Million-AID	ResNet-50	24M	0.26	52.6%	0.23	4.35	13.9%
CMID FT	Million-AID	ResNet-50	24M	*0.52*	47.2%	0.22	4.04	04.6%
CMID	Potsdam	Swin-B	88M	0.06	57.5%	0.27	4.38	15.8%
CMID FT	Potsdam	Swin-B	88M	0.04	58.4%	0.27	4.45	17.1%
CMID	Million-AID	Swin-B	88M	0.03	57.1%	0.27	4.62	16.9%
CMID FT	Million-AID	Swin-B	88M	0.03	57.2%	0.27	4.42	17.3%

Table 4.3: Performance of the self-supervised pretrained models on remote sensing images (RSI). FT stands for finetuned model. The best results of each category are shown in bold. The best results overall are shown underlined. Results that outperform the baseline, SCIPE, are shown in italics.

Chapter 5

Conclusions

The primary goal of this project was to extend the SCIPE pipeline and investigate the effect of different self-supervised pretraining paradigms on census-independent population estimation with high-resolution satellite imagery. The secondary objective of this study dissertation was to improve SCIPE's performance.

To achieve this research aims, we conducted extensive experimentation and evaluated the performance of SCIPE on 25 different pretraining configurations including pretraining with different learning signals, datasets of various domains, sample sizes, and spatial resolutions, and different backbone architectures of various capacities.

We observed that the three SimCLR contrastive models pretrained on very-high-resolution (0.1 - 2 m) remote sensing images achieved the best per-tile population estimations, despite the really small scale of their pretraining datasets (including two of which had less than 2,500 samples each). The TOV model whose shallow layers are pretrained on ImageNet and later on satellite images with self-supervised pretraining achieves the lowest percentage error when estimating per region (AggPE = 0.6%).

Moreover, we observed that when pretrained on natural images, ViT-B models perform better than ResNet-50 models. A supervised ViT-B model pretrained on ImageNet outperforms SCIPE on all evaluation metrics and achieves the second-best aggregate percentage error of 0.7%. On the other hand, SwinT and Masked Autoencoder models are the worst performers.

Future work on building large-scale unlabeled high-resolution RS datasets and using those for self-supervised pretraining with contrastive models could improve population estimates. In addition, the characteristics of the learned representations need to be further understood. Qualitative assessments of the features, such as t-SNE visualization and activation maps, can help.

Overall, this research has contributed to the first evaluation of representation learning with self-supervised pretraining on census-independent population estimation using high-resolution satellite images. The findings of this project make SCIPE even more sustainable by using label-free datasets for pretraining and completely avoiding the manual extraction of zero-population tiles.

Bibliography

Kumar Ayush, Burak Uzkent, Chenlin Meng, Kumar Tanmay, Marshall Burke, David Lobell, and Stefano Ermon. Geography-aware self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10181–10190, 2021.

Maksym Bondarenko, Patricia Jones, Douglas Leasure, Attila Lazar, and Andrew Tatem. Gridded population estimates disaggregated from mozambique's fourth general population and housing census (2017 census), version 1.1. 2020.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9650–9660, October 2021.

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

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020b.

Muskaan Chopra, Prakash Chandra Chhipa, Gopal Mengi, Varun Gupta, and Marcus Liwicki. Domain adaptable self-supervised representation learning on remote sensing satellite imagery. *arXiv preprint arXiv:2304.09874*, 2023.

Jian Ding, Nan Xue, Yang Long, Gui-Song Xia, and Qikai Lu. Learning roi transformer for oriented object detection in aerial images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2849–2858, 2019.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.

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

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

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

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

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

Ziyue Huang, Mingming Zhang, Yuan Gong, Qingjie Liu, and Yunhong Wang. Generic knowledge boosted pre-training for remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

Neal Jean, Sherrie Wang, Anshul Samar, George Azzari, David Lobell, and Stefano Ermon. Tile2vec: Unsupervised representation learning for spatially distributed data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3967–3974, 2019.

Liming Jiang, Bo Dai, Wayne Wu, and Chen Change Loy. Focal frequency loss for image reconstruction and synthesis. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13919–13929, 2021.

Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):4037–4058, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Alexander C Li, Alexei A Efros, and Deepak Pathak. Understanding collapse in non-contrastive siamese representation learning. In *European Conference on Computer Vision*, pages 490–505. Springer, 2022.

Catherine Linard, Marius Gilbert, Robert W Snow, Abdisalan M Noor, and Andrew J Tatem. Population distribution, settlement patterns and accessibility across africa in 2010. *PloS one*, 7(2):e31743, 2012.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

Yang Long, Gui-Song Xia, Shengyang Li, Wen Yang, Michael Ying Yang, Xiao Xiang Zhu, Liangpei Zhang, and Deren Li. On creating benchmark dataset for aerial image interpretation: Reviews, guidances, and million-aid. *IEEE Journal of selected topics in applied earth observations and remote sensing*, 14:4205–4230, 2021.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Oscar Manas, Alexandre Lacoste, Xavier Giró-i Nieto, David Vazquez, and Pau Rodriguez. Seasonal contrast: Unsupervised pre-training from uncurated remote sensing data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9414–9423, 2021.

Nando Metzger, John E Vargas-Muñoz, Rodrigo C Daudt, Benjamin Kellenberger, Thao Ton-That Whelan, Ferda Ofli, Muhammad Imran, Konrad Schindler, and Devis Tuia. Fine-grained population mapping from coarse census counts and open geodata. *Scientific Reports*, 12(1):20085, 2022.

Dilxat Muhtar, Xueliang Zhang, Pengfeng Xiao, Zhenshi Li, and Feng Gu. Cmid: A unified self-supervised learning framework for remote sensing image understanding. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–17, 2023.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

Isaac Neal, Sohan Seth, Gary Watmough, and Mamadou S Diallo. Census-independent population estimation using representation learning. *Scientific Reports*, 12(1):5185, 2022.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil

Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Xiaoman Qi, Panpan Zhu, Yuebin Wang, Liqiang Zhang, Junhuan Peng, Mengfan Wu, Jialong Chen, Xudong Zhao, Ning Zang, and P Takis Mathiopoulos. Mlrsnet: A multi-label high spatial resolution remote sensing dataset for semantic scene understanding. *ISPRS Journal of Photogrammetry and Remote Sensing*, 169:337–350, 2020.

Vladimir Risojević and Vladan Stojnić. Do we still need imagenet pre-training in remote sensing scene classification? *arXiv preprint arXiv:2111.03690*, 2021.

Caleb Robinson, Fred Hohman, and Bistra Dilkina. A deep learning approach for population estimation from satellite imagery. In *Proceedings of the 1st ACM SIGSPATIAL Workshop on Geospatial Humanities*, pages 47–54, 2017.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Spreading vectors for similarity search. *arXiv preprint arXiv:1806.03198*, 2018.

Xian Sun, Peijin Wang, Wanxuan Lu, Zicong Zhu, Xiaonan Lu, Qibin He, Junxi Li, Xuee Rong, Zhujun Yang, Hao Chang, et al. Ringmo: A remote sensing foundation model with masked image modeling. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–22, 2022.

Chao Tao, Ji Qi, Mingning Guo, Qing Zhu, and Haifeng Li. Self-supervised remote sensing feature learning: Learning paradigms, challenges, and future works. *IEEE Transactions on Geoscience and Remote Sensing*, 2023a.

Chao Tao, Ji Qi, Guo Zhang, Qing Zhu, Weipeng Lu, and Haifeng Li. Tov: The original vision model for optical remote sensing image understanding via self-supervised learning. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2023b.

Tobias G Tiecke, Xianming Liu, Amy Zhang, Andreas Gros, Nan Li, Gregory Yetman, Talip Kilic, Siobhan Murray, Brian Blankespoor, Espen B Prydz, et al. Mapping the world population one building at a time. *arXiv preprint arXiv:1712.05839*, 2017.

Xin-Yi Tong, Gui-Song Xia, Qikai Lu, Huanfeng Shen, Shengyang Li, Shucheng You, and Liangpei Zhang. Land-cover classification with high-resolution remote sensing images using transferable deep models. *Remote Sensing of Environment*, 237:111322, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Di Wang, Jing Zhang, Bo Du, Gui-Song Xia, and Dacheng Tao. An empirical study of remote sensing pretraining. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–20, 2022a.

Di Wang, Qiming Zhang, Yufei Xu, Jing Zhang, Bo Du, Dacheng Tao, and Liangpei

Zhang. Advancing plain vision transformer toward remote sensing foundation model. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–15, 2022b.

Yi Wang, Conrad M Albrecht, Nassim Ait Ali Braham, Lichao Mou, and Xiao Xiang Zhu. Self-supervised learning in remote sensing: A review. *arXiv preprint arXiv:2206.13188*, 2022c.

Yi Wang, Nassim Ait Ali Braham, Zhitong Xiong, Chenying Liu, Conrad M Albrecht, and Xiao Xiang Zhu. Ssl4eo-s12: A large-scale multimodal, multitemporal dataset for self-supervised learning in earth observation [software and data sets]. *IEEE Geoscience and Remote Sensing Magazine*, 11(3):98–106, 2023.

Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3974–3983, 2018.

Zhenda Xie, Yutong Lin, Zhuliang Yao, Zheng Zhang, Qi Dai, Yue Cao, and Han Hu. Self-supervised learning with swin transformers. *arXiv preprint arXiv:2105.04553*, 2021.

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

Yufei Xu, Qiming Zhang, Jing Zhang, and Dacheng Tao. Vitae: Vision transformer advanced by exploring intrinsic inductive bias. *Advances in neural information processing systems*, 34:28522–28535, 2021.

Yi Yang and Shawn Newsam. Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, pages 270–279, 2010.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International conference on machine learning*, pages 12310–12320. PMLR, 2021.

Qiming Zhang, Yufei Xu, Jing Zhang, and Dacheng Tao. Vitaev2: Vision transformer advanced by exploring inductive bias for image recognition and beyond. *International Journal of Computer Vision*, 131(5):1141–1162, 2023.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 649–666. Springer, 2016.

Tong Zhang, Peng Gao, Hao Dong, Yin Zhuang, Guanqun Wang, Wei Zhang, and He Chen. Consecutive pre-training: A knowledge transfer learning strategy with relevant unlabeled data for remote sensing domain. *Remote Sensing*, 14(22):5675, 2022.

Zhuo Zheng, Yanfei Zhong, Junjue Wang, and Ailong Ma. Foreground-aware relation network for geospatial object segmentation in high spatial resolution remote sensing imagery. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4096–4105, 2020.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021.

Appendix A

Performance with Zero-Population Tiles

Model	Dataset	Backbone	# Params	R^2	MeAPE	AdjMAPE	MeAE	AggPE
Supervised (NI)								
Swin-T	ImageNet	–	28M	0.12	49.0%	0.23	*3.76*	24.3%
Swin-T FT	ImageNet	–	28M	*0.47*	**41.9%**	**0.19**	***3.26***	13.8%
ViT-B/16	ImageNet	–	86M	0.12	51.2%	0.24	4.13	23.8%
ViT-B/16 FT	ImageNet	–	86M	***0.51***	48.6%	0.20	*3.65*	**08.9%**
Supervised (RSI)								
ResNet-50	Million-AID	–	24M	**-0.02**	55.3%	0.24	4.17	30.2%
ResNet-50 FT	Million-AID	–	24M	-0.08	59.8%	0.27	4.79	**13.6%**
Swin-T	Million-AID	–	28M	-0.04	**54.5%**	**0.23**	***3.89***	31.2%
Swin-T FT	Million-AID	–	28M	-0.03	54.7%	0.25	4.19	31.8%

Table A.1: Performance of the supervised pretrained models on natural images (NI) or remote sensing images (RSI) when the downstream dataset uses zero-population tiles. FT stands for finetuned model. The best results of each category are shown in bold. the best results overall are shown underlined. Results that outperform the baseline, SCIPE, are shown in italics.

Model	Dataset	Backbone	# Params	R^2	MeAPE	AdjMAPE	MeAE	AggPE
Self-Supervised (NI)								
SimCLR	ImageNet	ResNet-50	24M	0.06	53.9%	0.24	3.99	28.2%
SimCLR FT	ImageNet	ResNet-50	24M	-0.14	64.2%	0.26	4.41	29.4%
MoCo-v2	ImageNet	ResNet-50	24M	-0.09	59.8%	0.23	*3.68*	37.5%
MoCo-v2 FT	ImageNet	ResNet-50	24M	-0.05	62.6%	0.26	4.69	18.5%
MAE	ImageNet	ViT-B/16	86M	-0.04	53.7%	0.24	4.01	31.6%
MAE FT	ImageNet	ViT-B/16	86M	-0.08	60.2%	0.25	4.50	27.7%
DINO	ImageNet	ResNet-50	24M	0.00	55.5%	0.23	*3.90*	33.2%
DINO FT	ImageNet	ResNet-50	24M	-0.13	62.7%	0.27	4.60	20.9%
DINO	ImageNet	ViT-B/16	85M	-0.05	55.1%	0.26	3.98	30.9%
DINO FT	ImageNet	ViT-B/16	85M	-0.19	58.7%	0.25	4.16	26.7%
DINO-v2	LVD-142M	ViT-B/14	86M	0.13	53.5%	0.24	4.15	23.0%
DINO-v2 FT	LVD-142M	ViT-B/14	86M	**0.24**	**49.7%**	**0.21**	***3.64***	**12.0%**
MoBy	ImageNet	Swin-T	28M	-0.06	55.6%	0.24	4.18	31.8%
MoBy FT	ImageNet	Swin-T	28M	-0.03	54.4%	0.24	3.99	32.5%

Table A.2: Performance of the self-supervised pretrained models on natural images (NI) when the downstream dataset uses zero-population tiles. FT stands for finetuned model. The best results of each category are shown in bold. the best results overall are shown underlined. Results that outperform the baseline, SCIPE, are shown in italics.

Model	Dataset	Backbone	# Params	R^2	MeAPE	AdjMAPE	MeAE	AggPE
Self-Supervised (RSI)								
MoCo-v2	SSL4EO-S12	ResNet-50	24M	0.00	53.7%	0.23	*3.80*	25.7%
MoCo-v2 FT	SSL4EO-S12	ResNet-50	24M	*0.43*	47.5%	0.22	*3.61*	**07.1%**
MoCo-v2	Potsdam	ResNet-50	24M	0.04	52.2%	0.22	*3.71*	33.7%
MoCo-v2 FT	Potsdam	ResNet-50	24M	0.03	57.2%	0.26	4.72	24.5%
BYOL	Potsdam	ResNet-50	24M	0.20	53.3%	0.20	4.21	22.6%
BYOL FT	Potsdam	ResNet-50	24M	0.27	48.4%	0.23	4.16	14.9%
SimCLR	UC Merced	ResNet-50	24M	0.00	53.5%	0.23	4.02	31.4%
SimCLR FT	UC Merced	ResNet-50	24M	*0.48*	49.2%	0.20	*3.59*	12.9%
SimCLR	MLRSNet	ResNet-50	24M	0.08	53.4%	0.24	4.00	27.0%
SimCLR FT	MLRSNet	ResNet-50	24M	0.29	52.3%	0.22	*3.86*	18.3%
SimCLR	SIRI-WHU	ResNet-50	24M	-0.03	56.4%	0.25	4.15	28.2%
SimCLR FT	SIRI-WHU	ResNet-50	24M	0.31	49.0%	0.21	*3.59*	14.1%
MAE	Million-AID	ViT-B/16	100M	-0.07	58.4%	0.26	4.29	30.8%
MAE FT	Million-AID	ViT-B/16	100M	-0.02	58.5%	0.24	4.39	27.0%
MAE	Potsdam	ViT-B/16	86M	-0.01	55.5%	0.25	4.14	31.9%
MAE FT	Potsdam	ViT-B/16	86M	0.13	60.8%	0.24	4.17	20.3%
TOV	TOV-NI & TOV-RS	ResNet-50	24M	0.00	51.4%	0.24	3.93	26.1%
TOV FT	TOV-NI & TOV-RS	ResNet-50	24M	*0.42*	52.1%	0.21	3.91	12.6%
GeRSP	ImageNet & MillionAID	ResNet-50	24M	0.21	52.2%	0.22	*3.85*	22.6%
GeRSP FT	ImageNet & MillionAID	ResNet-50	24M	***0.52***	**46.0%**	**0.19**	*3.52*	15.2%
CMID	Potsdam	ResNet-50	24M	0.21	48.9%	0.21	3.54	25.9%
CMID FT	Potsdam	ResNet-50	24M	*0.48*	48.7%	**0.19**	***3.44***	11.0%
CMID	Million-AID	ResNet-50	24M	0.23	50.8%	0.23	4.03	21.9%
CMID FT	Million-AID	ResNet-50	24M	*0.50*	50.8%	0.22	3.97	09.3%
CMID	Potsdam	Swin-B	88M	-0.01	57.5%	0.26	4.14	30.9%
CMID FT	Potsdam	Swin-B	88M	-0.01	57.9%	0.25	*3.87*	30.0%
CMID	Million-AID	Swin-B	88M	-0.03	54.4%	0.23	4.27	29.9%
CMID FT	Million-AID	Swin-B	88M	-0.02	53.5%	0.24	4.18	30.0%

Table A.3: Performance of the self-supervised pretrained models on remote sensing images (RSI) when the downstream dataset uses zero-population tiles. FT stands for finetuned model. The best results of each category are shown in bold. the best results overall are shown underlined. Results that outperform the baseline, SCIPE, are shown in italics.