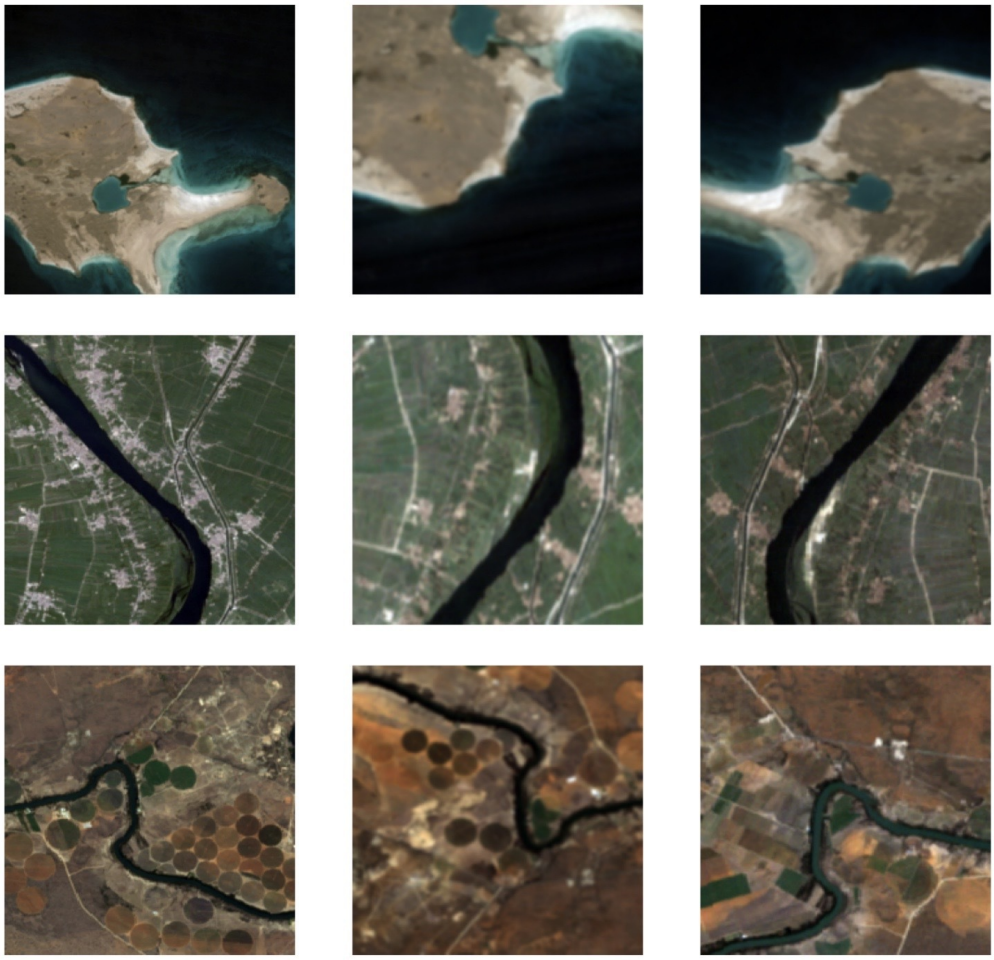


Predicting economic well-being in Africa using temporal satellite imagery and self-supervised learning

Jesper Strömberg, Benjamin Vinnerholt

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2022

Master's thesis 2022

Predicting economic well-being in Africa using temporal satellite imagery and self-supervised learning

Jesper Strömberg, Benjamin Vinnerholt

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2022

Predicting economic well-being in Africa using temporal satellite imagery and deep learning
Jesper Strömberg, Benjamin Vinnerholt

Supervisor: Adel Daoud, The Division of Data Science and Artificial Intelligence, Department of Computer Science and Engineering
Advisor: Mohammad Kakooei, The Division of Data Science and Artificial Intelligence, Department of Computer Science and Engineering
Examiner: Peter Damaschke, Department of Computer Science and Engineering

Master's Thesis 2022
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Augmented temporal satellite images of three locations in Africa.

Typeset in LaTeX
Gothenburg, Sweden 2022

Predicting economic well-being in Africa using temporal satellite imagery and deep learning
Jesper Strömberg, Benjamin Vinnerholt
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Accurate and reliable data on economic livelihoods remain scarce in the developing world, and major development agencies continue to study these outcomes and find the most effective means of assisting the impoverished. To accomplish this, wide and accurate local-level measurements of human well-being are necessary. Satellite imagery in the sense of measuring poverty has been proven a key data resource as it can fill in the resulting data gaps from scarce data. Prior research is based on predicting estimates of survey-based asset wealth from a time series of satellite images using convolutional neural networks. This work does not only implement these but also proposes a way of using self-supervised learning to improve the current state of the art models. Consequently, this work proposes novel training methods that exploit the spatio-temporal structure of remote sensing data. Through pre-training a network using contrastive learning with a MoCo framework and designated pretext tasks, one could increase the overall predictive performance in estimating poverty. The models were trained on surveys from 36 African countries and explained up to 66.4% of the variation in asset wealth at local-level locations, compared to 63.7% of an entirely supervised model.

Keywords: Deep learning, Remote sensing, Poverty prediction, Satellite imagery, MoCo, Temporal, CNN, ResNet, Self-supervised Learning, DHS, Contrastive Learning.

Acknowledgements

First and foremost, we would like thank our supervisors, Adel Daoud and Mohammad Kakooei for their invaluable input, guidance and feedback. We would also like to express our gratitude towards the rest of the team at the AI and Global Development Lab at Chalmers and Linköping University, who have helped us along the way by introducing us to concepts for deep learning on remote sensing tasks and providing feedback. Another extra thanks to Markus Pettersson, who conducted a thesis on a related subject in the previous year, who have helped us in familiarizing with the data and running models on a remote computing cluster.

Jesper Strömberg, Benjamin Vinnerholt, Gothenburg, August 2022

Contents

List of Figures

List of Tables

1

Introduction

About 900 million people, a third of the whole of Africa, are currently living in extreme poverty, i.e. they have an average consumption below 1.90$ per capita per day [1]. Poverty reduction is the focus of most developing-country governments today, and major development agencies continue to grapple with the most effective means of assisting the impoverished [2]. To accomplish this, accurate and local-level measurements of human well-being are necessary. These problems may then fall into a data challenge: a lack of poverty measurement data. One key resource comes from nationally representative consumption and asset wealth surveys stemming from *the Demographic Health and Survey* program [3]. However, they have limited repeated observations of individual locations making it difficult to measure local changes over time. Conducting surveys is also expensive and time-consuming. Satellite imagery data is one answer to cover these issues, which is freely available from projects such as Landsat. Instead of solely using the current survey information that is being collected, one can train machine learning algorithms to predict poverty, which would enable poverty estimation in locations where no measurements have been taken.

Machine learning offers methods for knowledge representation, prediction, and decision making by learning from complex data. One of the challenges of learning algorithms is to tailor predictive models from high-dimensional (image) data to lower-dimensional interpretable data to be used in other domains such as poverty research. Both these models and the data are of great value for other domains in which they can be used for causal inference and policy-making. For example, with such data, scholars can predict where to allocate monetary aid or the next intervention to alleviate poverty.

One branch of machine learning, *supervised learning*, which essentially means training a model by providing labeled examples, is limited by the manual data labelling required, in this case conducting surveys. The model would gain larger knowledge and predictive power with greater amounts of data provided, as in the case with on-the-ground surveys. However, conducting surveys is expensive and time-consuming. This work focuses instead on the application of *self-supervised learning* using satellite imagery, which is freely available in abundance. Self-supervised learning is essentially training a machine learning model without the need for labeled data. This area is currently rather unexplored in the field of poverty prediction and may open

up for better performing models despite the low fraction of labeled data.

One popular subfield in self-supervised learning is *contrastive learning*. Contrastive learning is based on computing the similarity between different data points, in our case satellite images. Instead of directly comparing images in their original state, contrastive learning methods derive feature representations of the data, which are of lower dimension than the original data, and compare their similarity in "feature space" instead. By defining which images are similar, e.g. by stating that images of locations that lie within close proximity to each other are similar, the model can learn to map the features extracted from similar images closely together in feature space, while mapping features from dissimilar images far away from each other in feature space. One new method within contrastive learning for images is *Momentum Contrast*, MoCo, which trains a model to be able to detect if two augmented versions of an image originating from the same image. MoCo has proven state of the art performance not only on the image classification task ImageNet [4, 5], but also for remote sensing tasks using satellite imagery [6, 7], making it suitable to be applied in the field of remote sensing poverty prediction.

Consider that contrastive learning is not related to poverty predictions per se, but with the tools that come with it, one could find meaningful ways of creating more powerful models concerning satellite imagery tasks.

This thesis is part of the research project “Observatory of Poverty” and is conducted in collaboration with The AI and Global Development Lab at Chalmers and Linköping University.

2

Background and Problem Statement

Predicting poverty is in large a missing data problem. Considering both the dimensions of time and location, poverty surveys can only go so far in providing us with measurements of poverty. With standard supervised learning, the model cannot be trained on regions for which we don't have survey data and will have to rely on the model's ability to extrapolate. With the ambition to be able to model poverty all over the world, it is of interest to investigate how self-supervised learning could be used as a pre-training step to transfer the knowledge of self-supervised contrastive learning algorithms to a supervised model in order to reduce the amount of data that needs to be collected to be able to model poverty accurately in Africa.

In recent years, numerous studies have been conducted to predict poverty by using satellite imagery [8, 9]. With the help of on-the-ground surveys, i.e. *Demographic and Health Surveys (DHS)*, they provide a representation of poverty for a specific place at a specific time.

2.1 Demographic and Health Surveys (DHS)

Ever since 1984, The Demographic and Health Surveys (DHS) program has provided assistance to the collection, analysis, and dissemination of more than 300 demographic and health surveys in over 90 countries. The main objective is to provide data for assisting planning, policy development decisions and program management [3]. The way of accomplishing this is by using several types of questionnaires. These collect data at a specific cluster of locations and each DHS survey could contain anything between 5 000 and 30 000 households and is conducted every 5 years. The cluster is sampled so that each round of surveys is representative of the whole country during that year [3].

Together these households act to be representative of the whole country during that year. DHS measures numerous different variables. This project will focus only on one of these: *The wealth index*. The idea of the wealth index is based on creating a less noisy measurement of a household's long-term economic well-being. The index is computed through a method called the International Wealth Index (IWI).

2.2 Temporality and Spatiality in Wealth Maps

The current state of the art in research about poverty prediction using deep learning is primarily derived from the works of Jean et al. and Yeh et al. [9, 8]. They have approached the poverty prediction task by focusing on the spatial parameter. Consequently, they have trained a model on one set of locations to predict the level of wealth in another location. Thus filling in the gaps for which no wealth asset surveys have been conducted.

Additionally, aside from the spatial parameter, surveys have been carried out for many decades, and it's safe to assume that the level of wealth for any given location may have changed during that time. Consequently, the missing data problem also depends on a time parameter. This means that the resulting model outputs a wealth value for each time step and location, i.e. latitude, longitude, and time. Ergo, the resulting model can serve as a powerful tool for social scientists and policymakers, to evaluate their actions and policies. For example, this can tell how wealth differs when affected by poverty targeting programs from international aid.

A master's thesis from the previous year, written by Pettersson et al., was also conducted on this topic and showed promising results using supervised learning models [10]. Additionally to improving on the works of Jean et al. and Yeh et al, a self-supervised learning model was also trained but showed relatively poor results due to time constraints on their work. The general idea was to train a self-supervised model to predict whether a daytime and nightlight image were from the same location, as the amount of nightlight can be indicative of the level of poverty [11, 12, 13]. However the author adds that the area was only lightly touched, and there is much more to explore.

This thesis is based on accepting that very challenge, to which recent research could provide necessary information to further apply self-supervised learning to poverty prediction.

2.3 Aims

The purpose of this project is to create machine learning models which can predict poverty levels at different geographical locations and at different points in time for countries in Africa. As preceded by the problem statement the objective is to solve a missing data problem. At the most basic instance, the models in the project take multi-spectral and nightlight satellite data from the time period 1990-2019 and output predictions for poverty levels.

The main objective is to explore whether self-supervised and semi-supervised learning algorithms can improve the results over entirely supervised models, which to the authors' knowledge, have not yet been extensively tested from a poverty prediction point of view.

2.4 Limitations

This project developed models used for predicting poverty in Africa and focused on improving predictive power over an entirely supervised model with self-supervised methods. The project did not aim to provide the best possible model for predicting poverty in and of itself, but rather to find a way to improve current models. Therefore, the results gathered were not be directly compared to previous studies, but rather to a baseline model.

The foremost limitation is that the predictions were limited to only African countries. The only data that was used used as wealth labels were those that come from DHS surveys. This followed from the reasoning that earlier papers have used these in the same matter, but also the lack of other data sources that explain poverty tracking in African countries. These surveys have been conducted in 36 of the 54 African countries, and thus, the model could only be trained and evaluated on these countries. The model could still be applied to predict in other countries, however, the performance cannot be accurately measured due to not having reliable poverty measurements to compare to. The surveys are freely available in the time period 1991-2019, limiting the model to be trained and evaluated in these years.

Regarding satellite data used, Landsat data has been used which has relatively poor resolution images but has greater time spans, i.e. a larger data set. This means that data can be used as far as from 1984. The usability of satellite imagery is also largely dependent on the availability of cloud-free images, i.e. images where the area of interest is not occluded by clouds. This puts further limitations on areas that can be used to train and evaluate the model.

2.5 Ethics

One ethical consideration relating to our data is providing confidentiality for the surveyed households. In order for their privacy to not be breached, survey responses have been aggregated into clusters on the scale of neighborhoods. Further, the geo-location of where the surveys were conducted has been offset by a randomized distance and direction [3]. In addition to this, the satellite images used in this project have a spatial resolution of 30 x 30 m/pixel. This, in conjunction with the measures taken by the DHS means any individual house or residence will be impossible to distinguish, and thus the privacy of the respondents will be respected.

3

Theory

This chapter is aimed at creating a basic understanding of the concepts and underlying models used in the project. This is done by introducing the data sources and concepts which are necessary to interpret the results of the thesis.

3.1 International Wealth Index (IWI)

For every DHS household questionnaire, there are several questions concerning the household's assets. In total, there are twelve different assets, consisting of seven consumer durables, three housing characteristics, and access to two public services. The durables include the possession of a TV, refrigerator, phone, bicycle, car, a cheap utensil, and an expensive utensil. The housing characteristics are the number of sleeping rooms, quality of the floor material, and quality of the toilet facility. The public services are access to clean water and electricity.

For each household asset of which information is collected, a weight is assigned to that asset through principal components analysis (PCA). Based on the first principal component, an IWI score can be computed. The scale ranges from 0 to 100, and if a household has all durables and has the highest quality housing and services, the wealth index would be assigned to 100. The IWI values are constructed by collecting DHS surveys held between the years of when satellite images have been captured across the many countries which have been surveyed during that time.

3.2 Satellite imagery

Multi-spectral remote sensing imagery can easily be attained from the freely available Google Earth Engine (GEE) [14]. The satellite imagery being used comes from the Landsat satellites 5, 7, and 8. The Landsat missions have since 1972, continuously acquired images of the Earth's land surface, providing remote sensing data to help managers and policymakers make informed decisions about natural resources and the environment. As of the year 2022, there have been a total of 9 Landsat satellites in orbit, with increasing quality. The earliest of the satellites used is the Landsat 5 which has orbited since 1984. Only since Landsat 5, multi-spectral imagery with a spatial resolution of 30x30 meters per pixel has been available. Thus putting a time limitation on which satellites to collect data from.

Additionally, nightlight data also contains relevant information regarding the prediction of human living standards [8, 10]. However, no single satellite captured nightlights for the entirety of the Landsat mission program, which meant that multiple data sources were required to fill the entire period[14]. From the 60:s until 2013, the Defense Meteorological Satellite Program (DMSP) captured nightlight data. From 2012 until now, nightlight data from the Visible Infrared Imaging Radiometer Suite (VIIRS) is available[15, 16]. The resolution of DMSP is 30 arc-second/pixel while VIIRS has a 15 arc-second/pixel resolution.

3.3 Deep learning

This section covers an overview of the use of *Convolutional Neural Networks* (CNN) in image data. Thereafter, *Residual Networks* is introduced, which is a particular class of CNN.

3.3.1 CNN

Convolutional Neural Networks (CNNs) is a type of network that is designed to process multi-array data, most commonly images. A typical architecture for a CNN can be seen in Figure 3.1. The core concept separating CNNs from other neural network architectures is the idea of convolutions, which are applied in the beginning of the network structure. Convolutional layers have filters which it applies to the image, and can be seen as extracting features and shapes from the image, such as lines and corners. Filters can be of different sizes, but some common sizes are 3x3, 5x5, and 7x7 pixels. These filters are swept across the image and for each stride a dot product between the weights of the filter and the input values is calculated and output into a feature map. An example of this can be seen seen in Figure 3.2. In the first layer, the input values are the pixel values of the input image. As can be seen in Figure 3.1, the convolution is first applied directly on the image of the robot. For the next convolutional layer, the input values are the feature map created from the previous convolutional layer. This means that if the first layer detects e.g. lines and corners, the second layers detects combinations of such features close to each other, essentially providing richer and more abstract representations of the image content the deeper into the network the values get.

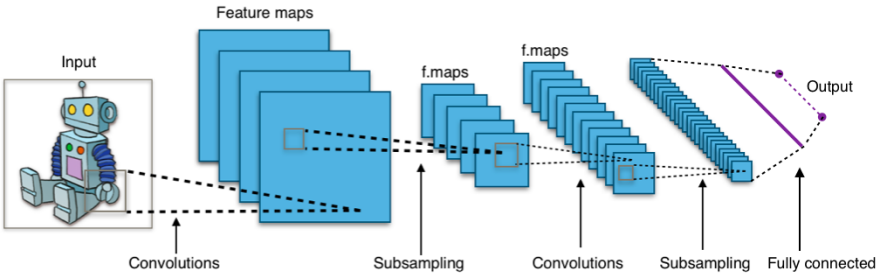


Figure 3.1: A typical architecture for a CNN [17].

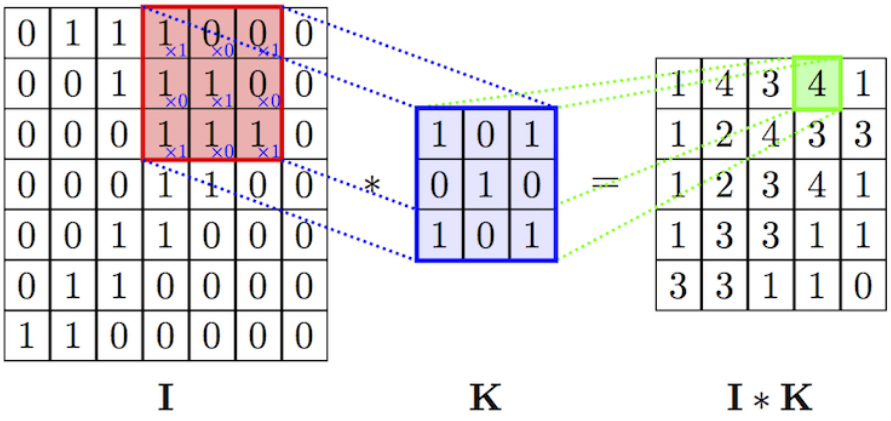


Figure 3.2: An example of a convolution [18]. I denotes the input values, K the filter, and $I \cdot K$ denotes the feature map created by the dot product.

Given that images are two dimensional arrays, often with three separate color channels as well, the amount of data coming out of a convolutional layer with several filters can become unwieldy, as well as containing weak signals carrying little information. CNNs solve this with *pooling layers*, which downscale the data by selecting only a single value within a specific region depending on some criterion. One common criterion is *max pooling*, which is simply selecting the maximum value within the region, and discarding the rest to reduce the size of the data. These regions can be of different size. An example of max pooling with a region size of 2x2 can be seen in Figure 3.3. As can be seen, for each 2x2 region, only the maximum value is selected and passed forward, to then be fed through the next layer in the CNN.

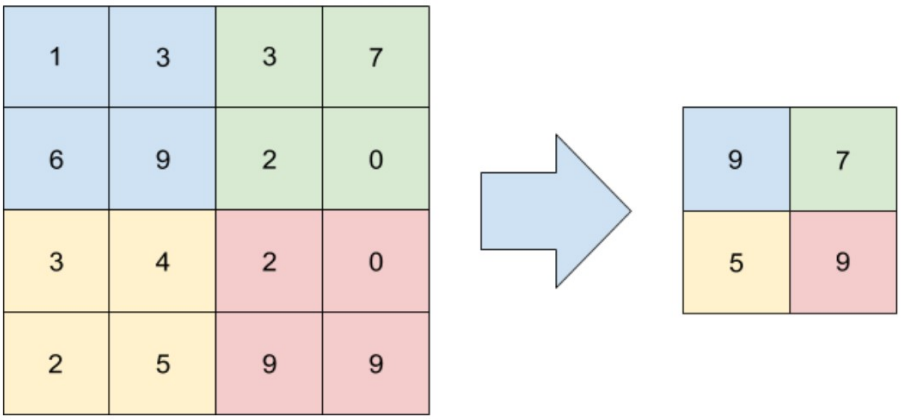


Figure 3.3: An example of max pooling

3.3.2 ResNet

Residual Networks, or ResNets, are a specific type of CNN. As previously stated, the deeper into a CNN the data gets, the more abstract and rich it becomes. Learning meaningful representations for any hard image task thus requires a deep network.

The problem with deep networks is that the amount of trainable parameters increases quickly, and simply creating a deeper network will not guarantee better results. Naturally, the amount of training needed for a deeper network is greater than that of a shallow one. This problem is handled in ResNets through *residual connections*. An example of residual connections can be seen in Figure 3.4. A residual connection essentially passes the input values x around a section of the network, as well as through it. On the other side of the network section, the output of the network section and the original input value x are added together. If the desired output after the residual connection is $f(x)$, the output of the network section thus instead needs to learn the mapping $f(x) - x$. This mapping is much faster and more stable to learn, allowing deeper networks to be trained. One such section with a residual connection is called a *residual block*, and the underlying structure of a ResNet is based on stacking these residual blocks after each other. Other than that the architecture is largely similar to a regular CNN.

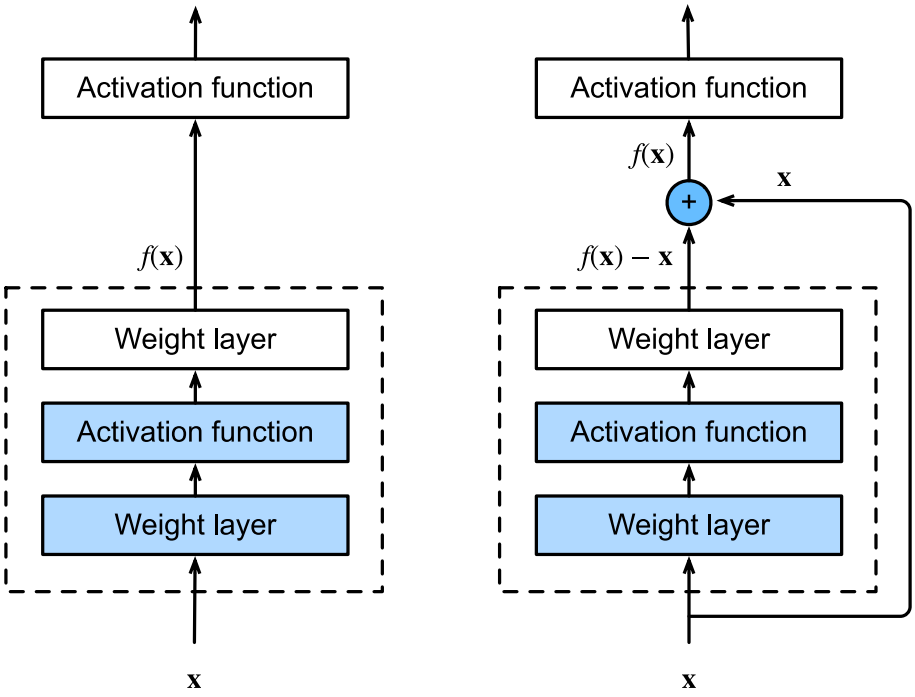


Figure 3.4: An example of a network without a residual connection (left), and a network with a residual connection (right) [19].

3.4 r^2 and RMSE

After retrieving predictions from the proposed model, an evaluation is made by measuring how well the model fits the output data through calculating the r^2-score and the RMSE, *root-mean-square error*, values. This was done using the equations presented in 3.1 and 3.2.

$$
\begin{aligned}
R^2 &= 1 - \frac{u}{v} \\
u &= \sum^{n} (y_{\text{pred}} - y_{\text{true}})^2 \\
v &= \sum^{n} (y_{\text{true}} - \bar{y}_{\text{true}})^2
\end{aligned}
\tag{3.1}
$$

$$
\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_i}{y_i} \right)} \tag{3.2}
$$

By those means, r^2 is a measure that represents the proportion of the variance for a response variable that's explained by an independent variable. RMSE explains instead how far the predicted values are from the true values. These metrics serve as an indication how well the model performs in predicting the response variable.

3.5 DBSCAN

DBSCAN, or Density-based spatial clustering of applications with noise, is a clustering algorithm based on density. It works, simply put, by combining data points within close proximity into clusters. DBSCAN has two main parameters, a minimum distance between data points to be considered in the same cluster, here denoted *min_dist*, and a minimum number of points to create a cluster, here denoted *min_pts*. Some clusters created by the DBSCAN algorithm can be seen in Figure 3.5. The points marked in blue are considered one cluster, and the points in green are another cluster. The gray points are not associated with any cluster and are considered outliers.

The algorithm initializes by selecting a random point, it then looks within a radius of *min_dist* to find nearby points and adds them to the group. If there are any nearby points, the same procedure of searching within *min_dist* for other points is applied such that their nearby points are also added to the group. This step is iterated until all points in the group have searched within *min_dist* of their location. If the total amount of points included in the group exceeds *min_pts*, then the group is considered a cluster, otherwise, all points in the group are considered outliers. This procedure is then repeated for another point that has not yet been visited until all points have been visited.

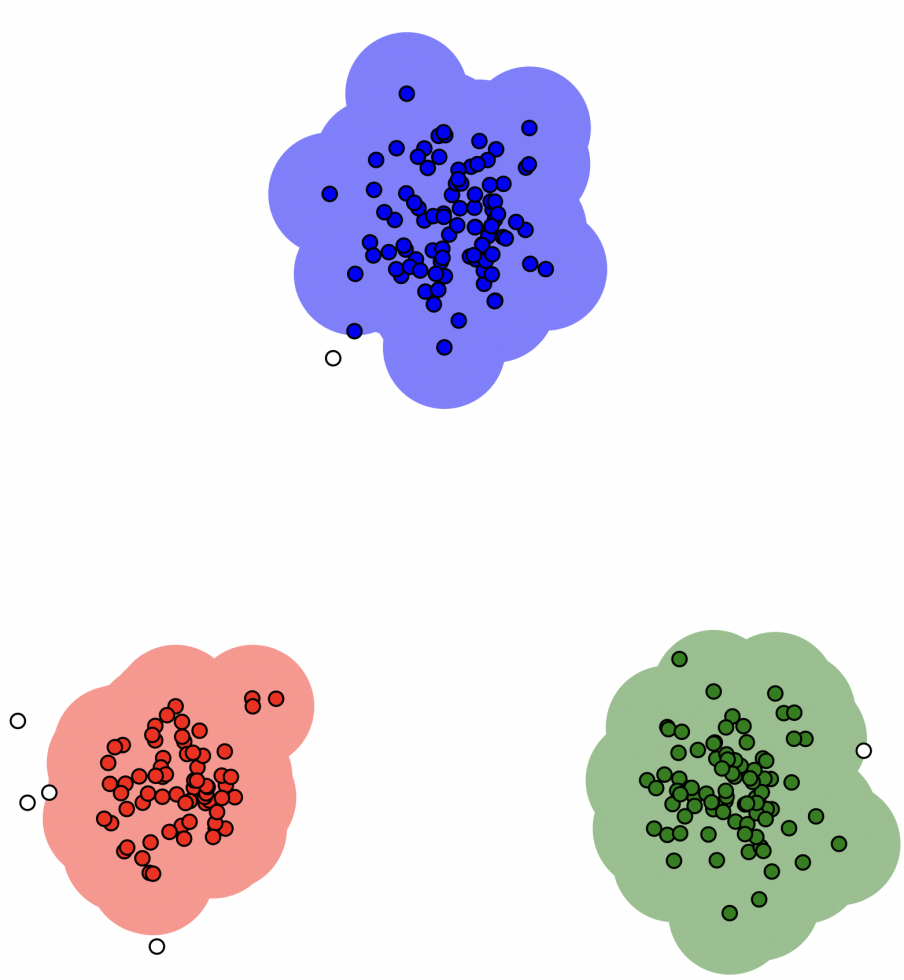


Figure 3.5: Clusters created from the DBSCAN algorithm [20].

3.6 Self-Supervised Learning

Self-supervised learning (SSL), in contrast to Supervised Learning, is not dependent on labelled inputs. This is important since manual annotation is an expensive and time consuming task. With SSL used in computer vision, one can learn visual features from large unlabelled data sets without the data being annotated. The cost of high-quality annotated data is a major bottleneck in the overall training process, which SSL doesn't need. The idea with SSL is to take advantage of the great pool of unlabelled data. A task that utilizes SSL is called a *pretext task*, and for each pretext task, there is a learning objective. Examples of pretext tasks could be to predict another part of an image, or given the grayscale version of the image, predict the colorized version. This means that the actual might be different from the final learning task, i.e. the downstream task such as poverty prediction.

For image recognition tasks in general, models pre-trained on ImageNet are the industry standard, but given that satellite data contains more channels (Landsat 9 has 11 bands, e.g. near-infrared) compared to ImageNet images (only RGB). This

means that there is no readily available pre-trained model for satellite imagery with more than three channels. The ImageNet images don't only differ in the amount of channels, as the subject of the images are objects, such as car, dog, cake or person, meaning that the representations learned by a model pre-trained on ImageNet might not be as meaningful when considering satellite imagery.

Since DHS surveys are mapping poverty over time, essentially labeled data, these can be used to train the model supervisedly after it has been trained with self-supervision. This is called semi-supervision and has been shown by Chen et al. to generally increase the performance of supervised models when pretrained with e.g. the model SimCLR [4], which is one of the popular methods based on *contrastive learning*. However, this has not yet been researched in poverty predictions in any way.

3.6.1 Contrastive Learning

Contrastive learning is a general class of pretext tasks that have been proven to be one of the most powerful approaches in self-supervised learning. In essence, contrastive learning makes it possible for the machine learning model to learn high-level feature representations of images. Contrastive learning is based on the idea of extracting feature representations from the data, and the aim is for similar images to have similar feature representations, i.e. lie close together in *feature space*. What makes two images similar is entirely defined by the user depending on the pretext task. For example, two images can be similar by being two different augmented versions of an image, or they can be similar by being from the same location.

The idea is not to find a final optimal performance for the pretext task, rather it revolves around the idea to find a reasonable learned representation with the expectation of carrying good semantic or structural meaning which could be beneficial to various downstream tasks. This means that the model can learn without any type of labels at hand for the downstream task.

The framework for which this project will consider is one first provided by He et al. called *momentum contrast*, MoCo [21], which is based on the same architecture as the model introduced by Chen et al., i.e. SimCLR [4].

MoCo-V2

MoCo is a state-of-the-art model based on using parallel data augmentations to learn visual representations. The model has never, to the authors' knowledge, been implemented from a poverty research point of view. The framework is based on contrastive learning, i.e. creating similar feature representations of images that are similar. For each image in the data set, data augmentation is performed randomly, see for example Figure 3.6. The augmentation could mean anything that is altering the image such as cropping, resizing, color distortion, blurring etc.

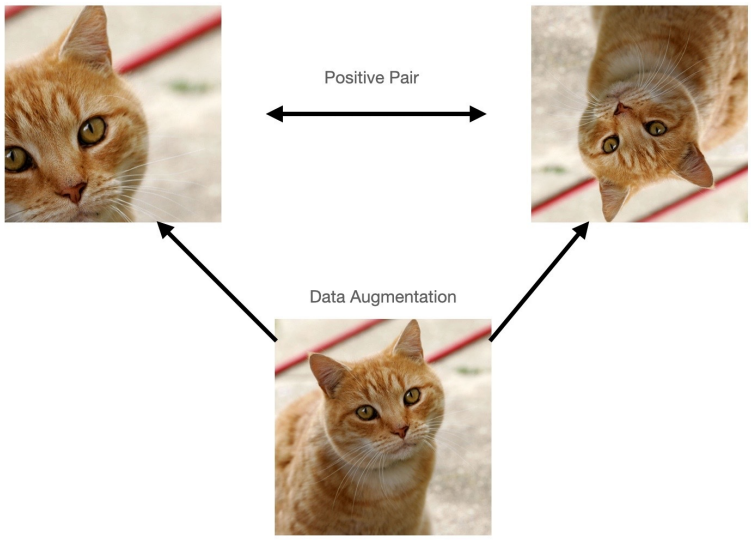


Figure 3.6: Showcase of image being augmented through zooming and flipping. These images are de facto a positive pair since they are from the same image.

The model knows that these two images are similar since they essentially are different versions of the same image, but needs to map them to similar feature representations. In order to accomplish this, both images are fed into separate deep feature extracting models, e.g. a CNN such as ResNet. The output from the CNN is then fed through a set of dense layers, called a *projection head*, which projects the features extracted from the CNN into the *latent feature space*. Thus, vector representations for each image are created, i.e. encoding the images. The goal is to train the model to output similar representations for similar images. The typical schema for this task can be found in Figure 3.7.

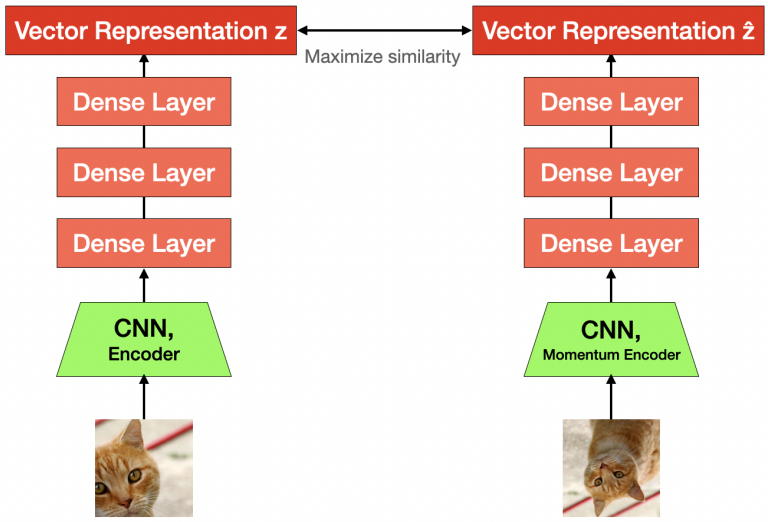


Figure 3.7: Architecture from fed image to vector representation.

Finally, when the two vector representations are produced, the similarity between the two can be quantified. How this works out is by computing the *cosine similarity* between the vectors. This basically means that when the angle between the vectors is close to 0, the similarity is high. Aside from calculating the cosine similarity, in order for the model to learn and improve performance, a loss function has to be used which can be minimized. The contrastive loss function InfoNCE is used in the MoCo-V2 framework [22]. The first step is based on the *softmax function*, and calculates the probability that the second image is the most similar image to the first image out of all other images. Figure 3.8 depicts a comprehensible version of this function.

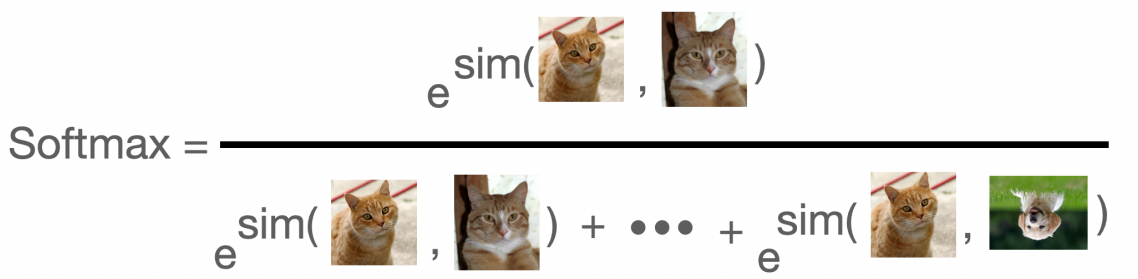


Figure 3.8: The softmax function, illustrated by using images, which is the first step of calculating the InfoNCE, a type of cross-entropy loss.

In a more mathematical sense, the contrastive loss function InfoNCE can be formulated, as in Equation 3.3. Note also, in a second step, that in order to minimize the contrastive loss, one intuitively has to maximize the probability, hence the negative logarithm.

$$L_z = -\log \frac{\exp(z \cdot \hat{z}/\lambda)}{\exp(z \cdot \hat{z}/\lambda) + \sum_{j=1}^{N} \exp(z \cdot k_j/\lambda)}. \tag{3.3}$$

Here, z and $\hat{z}$ corresponds to the query and key representations obtained by the two augmented views of a certain remote sensing image, i.e. the vector representations of the two images. Contrastive loss measures the similarity between the query z and the keys k_j where $\hat{z}$ is the first key representation. The symbol λ represents the temperature hyper-parameter. The number N of negative samples is extracted from a dictionary of representations built as a queue.

The loss function aims to make a query similar to its positive key and dissimilar to the negative key. Later two separate encoders are made for both the query and the key. Then the InfoNCE contrastive loss with temperature λ is used over one positive and $N-1$ negative samples. The dictionary enables the framework to reuse representations.

Architecture

MoCo proposes a momentum-based update with a momentum coefficient $m \in [0, 1)$. This is a novel feature of MoCo compared to previous contrastive learning methods. The purpose is to maintain consistency among representations in the queue. Per using a queue of representations, one cannot rely on back-propagation to update the key encoder called f_k, instead it is updated by the momentum update rule. The key encoder f_k has parameters θ_k and the momentum encoder(query encoder) f_q has parameters θ_q. The momentum update can then be expressed by the following scheme:

$$\theta_k \leftarrow m\theta_k + (1 - m)\theta_q, \tag{3.4}$$

In this manner, the parameters of the momentum encoder are updated in consistency with the encoder. This was shown to prove good performance according to He et al. [21].

4

Previous Works

4.1 Combining satellite imagery and machine learning to predict poverty

Jean et al. were among the first that attempted to develop deep learning models for detecting and predicting poverty using machine learning and satellite imagery. Through collecting poverty measurements from African DHS surveys and satellite images across the locations where households have been surveyed, independent models were trained on separate countries. Prior to their work, previous projects have shown that there exists a correlation between the intensity of nightlight in satellite imagery and levels of economic well-being [11, 12, 13]. However, the authors claimed that nightlights are unable to detect economic well-being in regions that lie below the international poverty line, i.e. an average consumption below $1.90 per capita per day.

Consequently, machine learning approach based on the technique of *transfer learning* is proposed. Instead of solely using nightlight images to make predictions, the authors train a CNN to predict nighttime luminosity levels from daylight images as a pre-text task. These learned representations, could for instance serve as indicators for economic well-being, such as urban areas, rural areas, water and roads. Next, they take advantage of these newly learned representations and the CNN is adapted to be trained to estimate night light intensities from daylight images as a pretext task. The training data is large since the amount of daylight and nightlight data is plentiful and spans the whole continent, which could serve useful when creating a better pre-trained CNN for the downstream supervised task. Thus, the last step involves combining the acquired DHS survey data and image feature representations extracted by the CNN from the daylight imagery. Finally, a ridge regression model is later trained to be able to estimate the poverty indicators. The indicators being used are the consumption expenditure and asset wealth.

The model was trained and evaluated on five different African countries: Nigeria, Malawi, Rwanda, Tanzania and Uganda. The evaluation was done by calculating the r^2-value, which states what fraction of the variance of the data can be explained by the model, for the best fitted line for the labels and the predicted values. Jean et

al. reported an r^2-value of 0.75 in-country and for countries not within the training locations showed a r^2-value of 0.56 [9].

Although, for every time the CNN is updated, it's never updated based on the model predictions. Rather, the lower-dimensional vector representations of each image is updated, which consequently affects the model predictions. The model aims to find features that actually are relevant to poverty predictions to gain deeper and more reasonable feature representations.

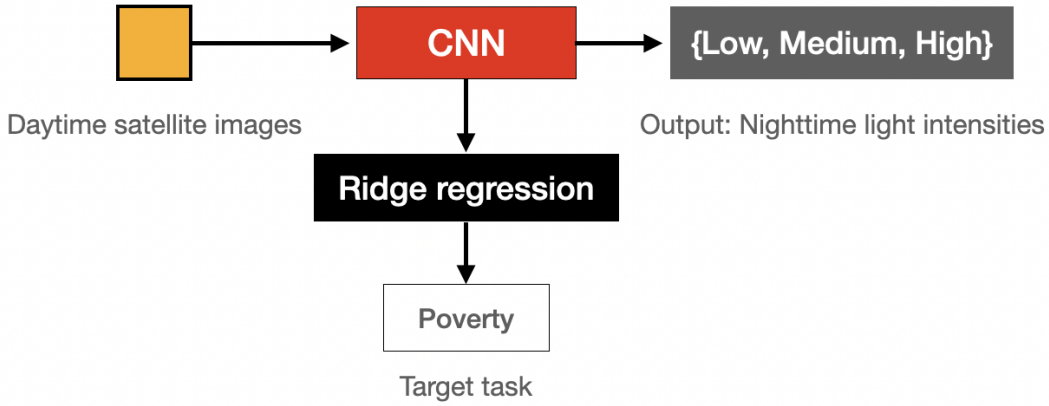


Figure 4.1: Model pipeline used by Jean et al. [9]. Images summarized as low-dimensional feature vectors. Each sample of daylight satellite images is fed individually through the CNN to train on a nightlight classification task. The intuition is that features representing nightlight intensity should be correlated with features representing poverty.

4.2 Using publicly available satellite imagery and deep learning to understand economic well-being in Africa

Jean et al. were one of the first to incorporate nightlight images in a pretext task for transfer learning. Yeh et al. instead introduced nightlight images as input to the final model [9, 8]. Yeh et al. created a model consisting of two ResNet:s that were specifically trained on either nightlight or multispectral images. The resulting feature vector outputs from both these layers are then jointly fed through a ridge regression layer, which is trained to predict the asset wealth index. The idea behind this can be seen in Figure 4.2.

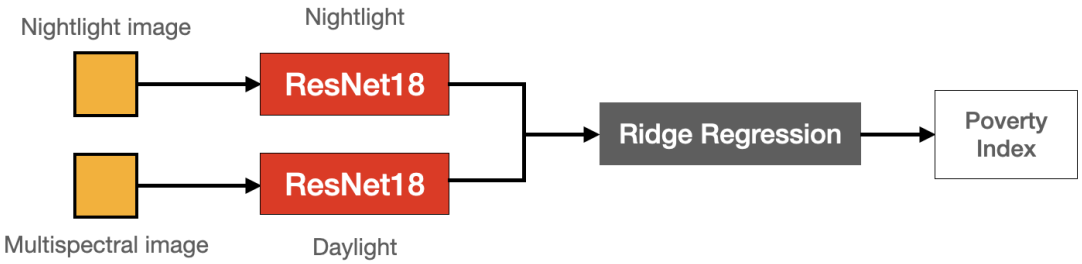


Figure 4.2: Basic architecture showcasing Yeh et al.'s approach to predicting the asset wealth index. Two ResNet18:s are trained simultaneously on both daylight multispectral and nightlight images. The resulting output of the ResNets are concatenated and fed through a Ridge regression layer that predicts asset wealth index.

Both the nightlight and multispectral models were evaluated individually and seemed to perform just as well as they did jointly, with the night light model being marginally better. The model input for Yeh et al. differed however from Jean et al. since they collected images from the Landsat program and not the Google Static API. This was done despite the lower spatial resolution, but having the benefits of being timestamped in a such way that the correct images could be captured close in time to when the actual survey took place.

Landsat imagery has a spatial resolution that covers 30×30 m and ResNet-18 has an input size of 224×224. This resulted in an image with a side length of 6.72 km. This created a problem for which some cluster points within rural locations could appear entirely outside the captured image because the displacement of each DHS survey has a 10 km radius. The benefits however of using images with smaller area coverage, means that the model would be less noisy since each image lies in close proximity to the DHS survey location, e.g. areas for which both wealthier and poorer households exist. Yeh et al. created 3-year median composite images for each survey location. With the use of Landsat, more images could be captured. Around 19 669 villages across 23 countries were incorporated into the model. Yeh et al. reported a r^2 value of 0.70, and 0.67 for held-out countries.

4.3 Geography-Aware Self-Supervised Learning

In contrast to the works of Jean et al. [9] and Yeh et al. [8], the paper by Ayush et al [6] does not deal with poverty predictions. Instead, their work explores the application of contrastive learning to remote sensing data sets, where there is a lack of labelled data but unlabelled data is often abundant. Ayush et al. proposed a way of extending the state of the art contrastive framework MoCo. As opposed to typical computer vision images, remote sensing data are geo-located and might even provide multiple images of the same location over time. Since contrastive learning methods is benefited from having image representations close in space, there could be an argument for using remote sensing images that are semantically similar from the

same locations but in different points in time. Consequently, Ayush et al. propose a method for using *temporal positive pairs* from spatially aligned images over time in contrast to typical computer vision images where different views of the same image act as a positive pair. A demonstration of temporal positive pairs can be seen in Figure 6.1. As can be seen, several roads and buildings have been constructed between the times the images are captured.

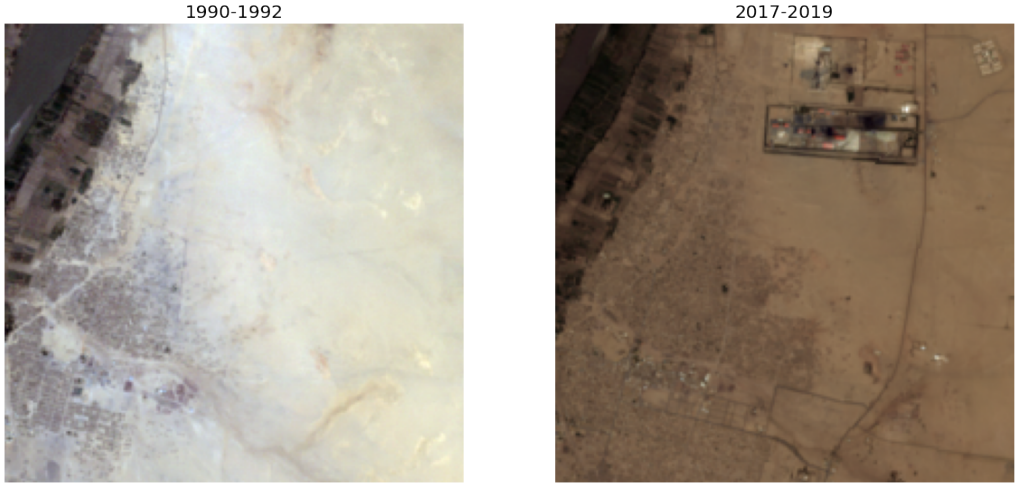


Figure 4.3: Two spatially aligned images from different times - temporal positive pair.

The authors claim that with a temporal pairs model using remote sensing data, one allows the representations to be invariant over time, e.g. due to seasonality. This results in a pretext task more inclined toward spatial variation, such as object detection or semantic segmentation. In the study, they perform unsupervised representation learning on fMoW and GeoImageNet data sets. fMoW, *functional Map of the World*, is a data set consisting of high spatial resolution satellite images ranging from 10 m to 60 m in spatial resolution. The pre-trained models are later evaluated on a variety of downstream tasks such as image recognition, object detection and semantic segmentation benchmarks. The models are also compared with a supervised learning benchmark which also runs on the same backbone CNN, i.e. in this case ResNet50. Analysis and experiments that relate to this thesis are the results coming from the fMoW data set since it's the only data set that includes satellite images of the same locations across different periods of time.

The schematic overview for the *temporal pairs* approach can be seen in Figure 4.4. Note that the only difference compared to the original MoCo-v2 framework lies in the use of temporal pairs. The two spatially aligned images is augmented to v, v' respectively. The resulting image representations are is z, z' when being fed to the query and key encoders f_q and f_k. Together these visual representations are compared through computing the contrastive loss, InfoNCE, and the query encoder f_q is updated using momentum update as in the same way as explained in the theory section 3.6.1.

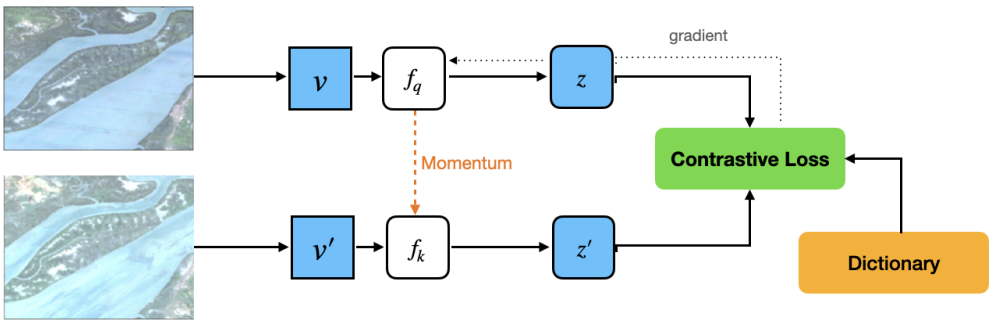


Figure 4.4: Schematic overview of MoCo-v2 framework with use of temporal positive pairs. v and v' represents the augmented views of the same image pair.

The evaluating standard was computing the f1-score and accuracy measurements. The proposed method was shown to outperform the supervised benchmark on all tasks when evaluating the accuracy metric. These include image recognition ($\sim$ +3%), single image classification ($\sim$ +2%) and classifying temporal image classification ($\sim$ +2%).

5

Data Management

This section describes the steps taken to retrieve and pre-process survey and satellite data, and also covers some data exploration for the DHS surveys. Further, the procedures for setting up the data for evaluation of the models are also defined.

5.1 Model Input and satellite image pre-processing

The image samples for locations were gathered from the Google Earth Engine API. The Google Earth Engine (GEE) is a cloud computing platform used for processing earth observations, such as satellite imagery and climate data [14]. In choosing a satellite project to collect images from, there are plenty of pros and cons depending on what type of satellite one wishes to use. For example, the satellite project Sentinel has a resolution of 10x10 meters/pixel but has only been in orbit since 2014, and therefore the temporal dimension is lacking. Landsat, on the other hand, has been in orbit for over 30 years but has a resolution of 30x30 meters/pixel. Given that the DHS surveys have been conducted since 1990, Landsat was chosen to be able to make use of the most amount of surveys, even though it has a significantly lower spatial resolution.

All image data were collected from the Landsat 5, 7 and 8 satellites, and with that comes necessary pre-processing. Landsat data often requires adjustments prior to analysis to account for sensor, solar, atmospheric and topographic effects [23]. Every satellite has its own configuration, and the pre-processing of a sensor reading from a satellite consists of several steps. The most common steps include georeferencing, co-registration, conversion to radiance, solar correction, atmospheric correction, topographic correction and relative correction [23]. Every step can be seen as each representation closer to the surface level. In this thesis, where one attempts to detect changes and utilizes images from multiple sensors across long time periods, a uniform distribution is needed. Therefore the surface level reflectance is preferred.

In this project two different data sets were used, one for the purpose of the supervised downstream task while the other more dense data set was used for the self-supervised pretext task. Both datasets are captured from the same satellites and have the same band set-up which spans over the same amount of years, i.e. between 1990 and 2019.

The satellite bands can be seen in table 5.1, which are the same used by Yeh et al. [8].

Type of Band	Description	Wavelength
BLUE	Blue	0.45 - 0.52 μm
GREEN	Green	0.52 - 0.60 μm
RED	Red	0.63 - 0.69 μm
NIR	Near infrared	0.77 - 0.90 μm
SWIR1	Shortwave infrared 1	1.55 - 1.75 μm
SWIR2	Shortwave infrared 2	2.08 - 2.35 μm
TEMP1	Brightness temperature (Kelvin)	10.40 - 12.50 μm

Table 5.1: The multi-spectral channels that were used in both the downstream supervised task as well as the self-supervised pretext task.

Beyond the multi-spectral bands that were used, the model also includes nightlight satellite imagery. From previous works of Jean. et al, this information has been proven useful when estimating asset wealth [9]. Since no single satellite collected nightlight data over the entire time period of 1990 to 2019, two different nightlight image collections were used to pair up with the multi-spectral data. Before 2015, nightlight imagery from the Defense Meteorological Satellite Program (DMSP) [15] was used while after 2015, imagery from Visible Infrared Imaging Radiometer Suite (VIIRS) [16] was used. The DMSP data has a spatial resolution of 30 arcseconds/pixel corresponding to roughly 926m along the equator, and VIIRS has a spatial resolution of 15 arcseconds, which corresponds to 463m.

Every image used as model input has the shape of 224×224 pixels, which is the standard input size of ResNet models, which are used in this project. Along with the 30 m/pixel spatial resolution of the multispectral bands, this corresponds to an image covering an area of 6.72×6.72 km. The nightlight band is upscaled to also be 224×224 pixels, while still covering the same area as the multispectral bands.

Additionally, in order to reduce the effects of seasonality, as well as be able to provide as large of a portion of the map as possible with cloud-free images within each timeframe, the satellite images are further composed into three-year composites. This has also been done in previous work such as Yeh et. al. and Pettersson et al. [8, 10]. A median image is calculated for each period of 3 years, such that the first set encompasses the timespan 1990-1992, while the last one covers 2017-2019. Therefore, 10 images were created per scene, and each image covers 8 bands which results in 80 bands per satellite image patch.

5.2 Data sets per task

In the following sections, the distinction of model input between the supervised downstream task and the self-supervised pretext task is presented.

5.2.1 Supervised Downstream Task

For the supervised downstream task, one image is collected for every location where a DHS survey has been conducted. As can be seen from Figure 5.1, which depicts the location of survey points, images are gathered from 36 of the 54 countries across the African continent. This means that not all countries are included, such as Botswana, Algeria and Sudan to name a few. Each image file collected contains one image for all of the 10 three-year timespans, however only the timespans where surveys have been conducted in the area of the image are used for supervised training. This means that the supervised data set contains 57 195 images, as many as the amount of surveyed clusters.

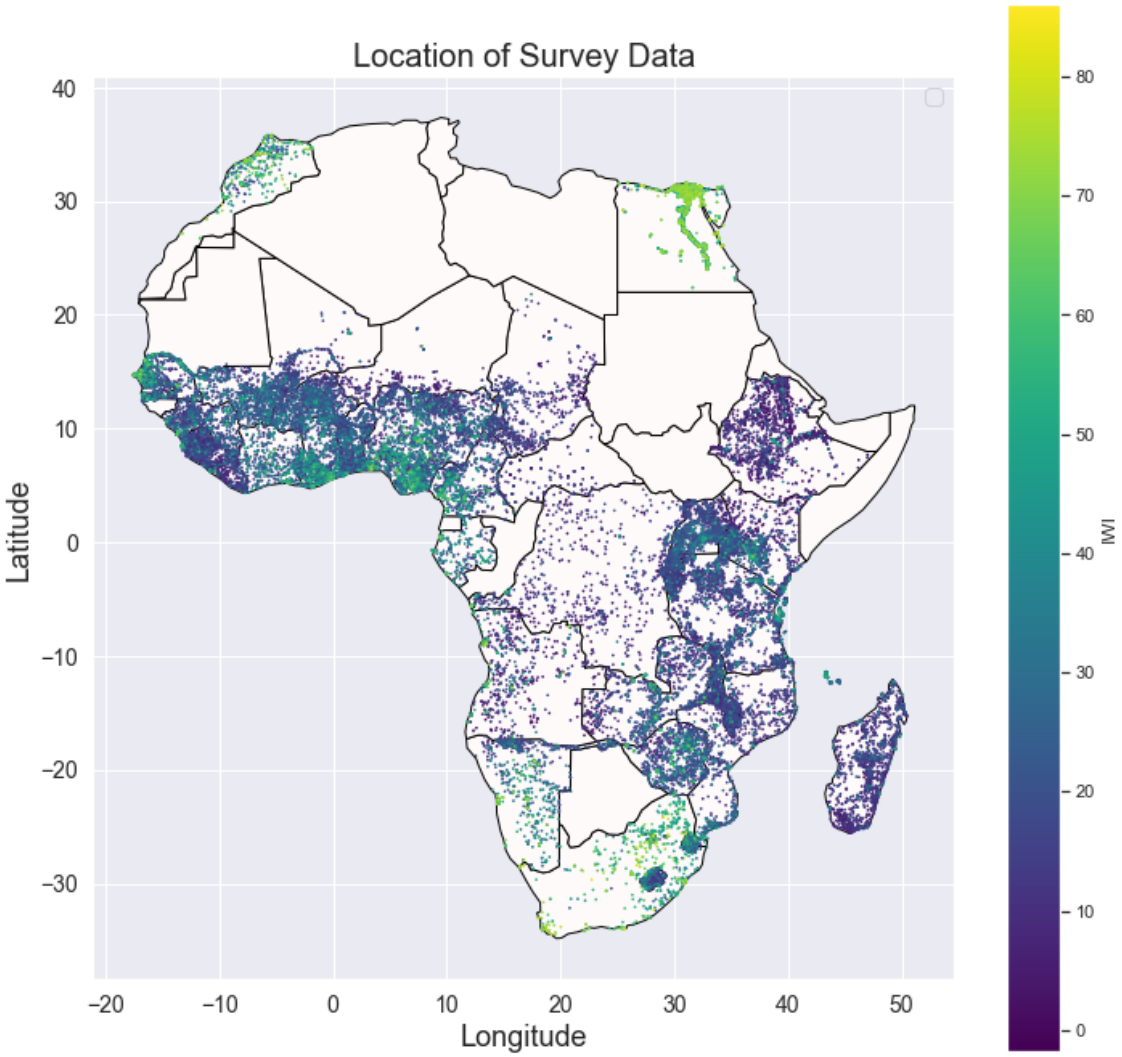


Figure 5.1: Wealth asset index distribution containing 57 195 DHS clusters spanning 36 countries surveyed between 1991 to 2019. The color of a point corresponds to wealth asset index of the location. The lighter the point, the generally wealthier the location.

5.2.2 Self-Supervised Pretext Tasks

In contrast to the supervised downstream task, the model input for self-supervision does not depend on any DHS clusters, i.e. does not need to be in close proximity to any surveyed households. Each and every captured image belong to a range of different areas and cover all of the 54 African countries. The images are however chosen with regard to population, which means that images are only collected for inhabited areas. The reasoning for this is that the data for the self-supervised model should be of similar areas and domains as the supervised data, i.e. based on inhabited areas, such that the knowledge gained from training self-supervisedly is applicable when transferred to the supervised model. Learning to extract features from images containing no human settlements, such as strictly desert, would presumably not be particularly informative for predicting poverty, a value based on human living conditions. The areas chosen can be seen in Figure 5.2.

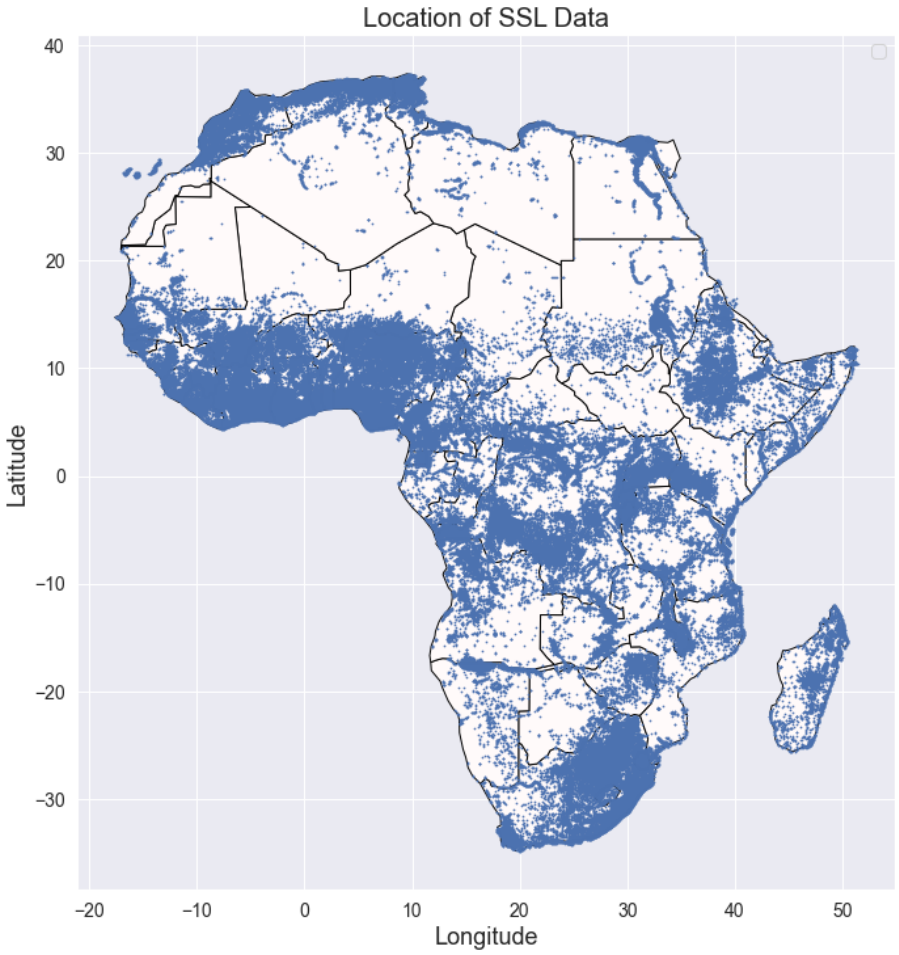


Figure 5.2: The figure showcases the areas for which images have been captured between 1990-2019. In contrast to the supervised downstream task, the locations are not based on DHS surveys.

Note that this data set also contains images from countries that are missing in the

supervised data set, with the ambition to aid the generalization of the model to the entirety of the African continent. In total, the SSL data set contains images of 136 963 locations, which is more than double that of the supervised data set. Not only that, but similarly to the data for the supervised task, each image file contains one image for each of the 10 three-year timespans between 1990-2019. Since these images are not dependent on any timestamped event such as the year a survey has been conducted in, which the data for the supervised task is, all 10 timespans can be used, meaning the self-supervised data set contains a total of 1 369 630 distinct images, almost 24 times more than the supervised data set.

5.3 Construction of wealth asset index

Similar to Pettersson et al. [10], the wealth asset index is computed using several constituents from DHS surveys. 36 countries were surveyed in Africa and the asset wealth index was computed using Smits et al.'s method which was described in section 2.1 [24]. The values of the wealth asset indices range from -1.41 to 86.07.

Every DHS survey, there conveys information about latitude and longitude coordinates for each cluster. In this context, a cluster defines the groupings of households that participated in the survey. By those means, the wealth asset indices were averaged across those households that belong to that certain cluster. However, the DHS program states that in order to ensure respondent confidentiality, a displacement has been applied for the latitude/longitude positions. This means that for urban clusters, there contains a minimum of 0 and a maximum of 2 kilometers error. Additionally, rural clusters contain a minimum of 0 and a maximum of 5 kilometers with a further 1% of the rural clusters displaced at a minimum of 0 and a maximum of 10 kilometers. The displacements are restricted in such a way that the points have to stay within the country and DHS survey region [3].

The clusters may contain anywhere anything between 1 to 156 households, but, in the same manner, as Yeh et al.[8], the wealth asset indices were not weighted on the number of households or household members in the cluster. The idea is to model the average wealth in an area, not the total wealth. Weighting the wealth asset index on the number of respondents would cause dense areas with lower average wealth to be modeled as more wealthy than sparsely populated with high average wealth, which is not desired. If any clusters had invalid locations (e.g. having the image contain only water) or belonged to areas where no satellite images were captured, they were later removed. In total, 57 195 valid clusters were used for analysis in this project.

5.4 Exploration of DHS Surveys

Figure 5.1 represents the wealth asset index distribution of all available DHS clusters in Africa. A total of 57 195 DHS surveys are shown, spanning 36 countries between 1991 and 2019. What can be noticed is that Central Africa and Madagascar suffer

from the lowest IWI:s across the whole continent. However northern Africa and also locations close to shore seem to be generally wealthier.

Figure 7.4 depicts the distribution of rural and urban clusters, which unveils that there are generally more rural than urban clusters on the country level.

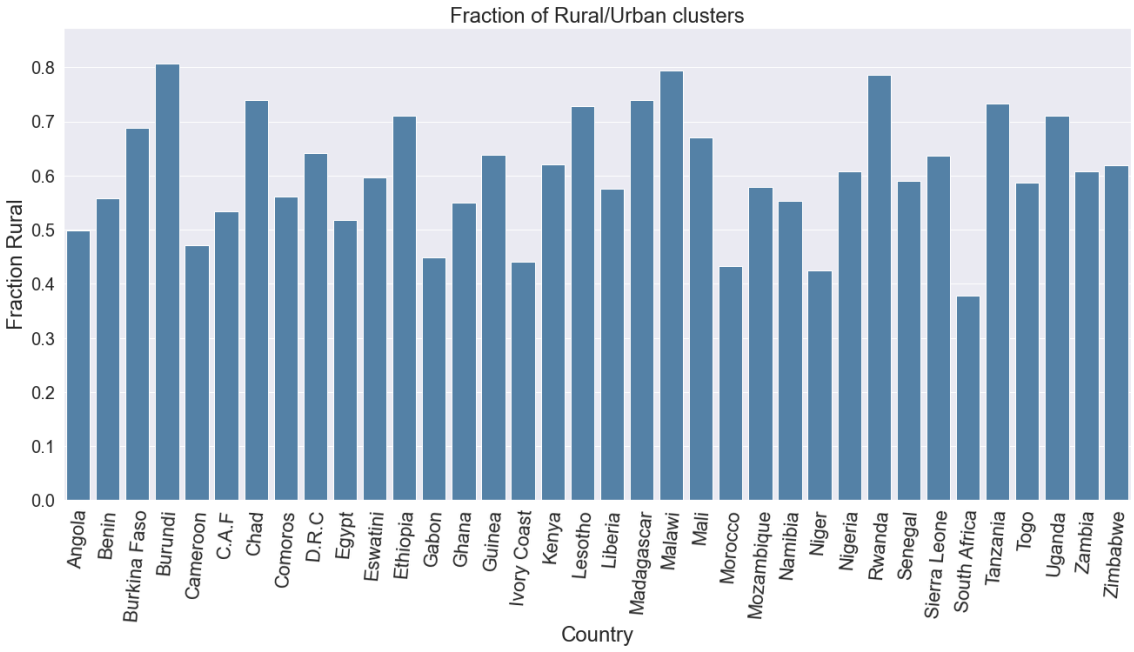


Figure 5.3: Distribution of rural clusters across the years 1990-2019. The distribution varies between 0.4 and 0.8.

Table 5.2 depicts the means and standard deviations of IWI across rural and urban clusters. As can be seen, rural clusters are more common than urban clusters, and the mean IWI of urban clusters is almost double that of rural clusters.

Urban/rural	Number of clusters	Mean IWI	Standard deviation IWI
Urban	21609	46.45	16.84
Rural	35586	23.50	15.39

Table 5.2: Number of points, and means and standard deviations of IWI for urban and rural clusters.

The number of clusters per country is shown in Figure 5.4, where it can be seen that some countries have been surveyed significantly more than others, e.g. Egypt. Note also that the median of the wealth asset index in Egypt is 62.07, compared to the rest of the surveyed countries having a mean of 22.77. By those means, the far more surveyed country Egypt, skews the distribution to a far more wealthier such. The number of clusters per year is as well shown in Figure 5.5. Here, it's clear to see that far more surveys has been done during some years, more prominently during

the latest few years. During the year of 1990 and 2002, no surveys has been done at all.

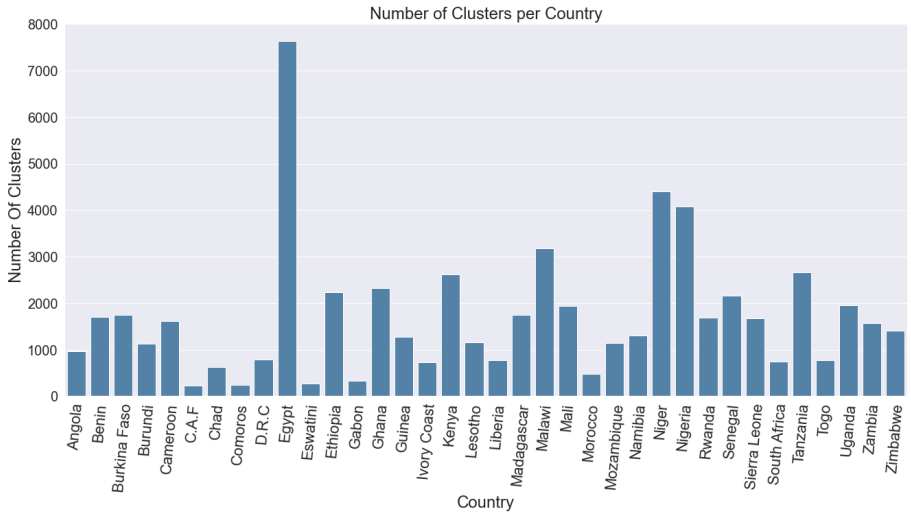


Figure 5.4: The Figure represents the number of clusters created for each country. It can be seen that some countries has been far more surveyed than others, and that Egypt has the largest portion of those.

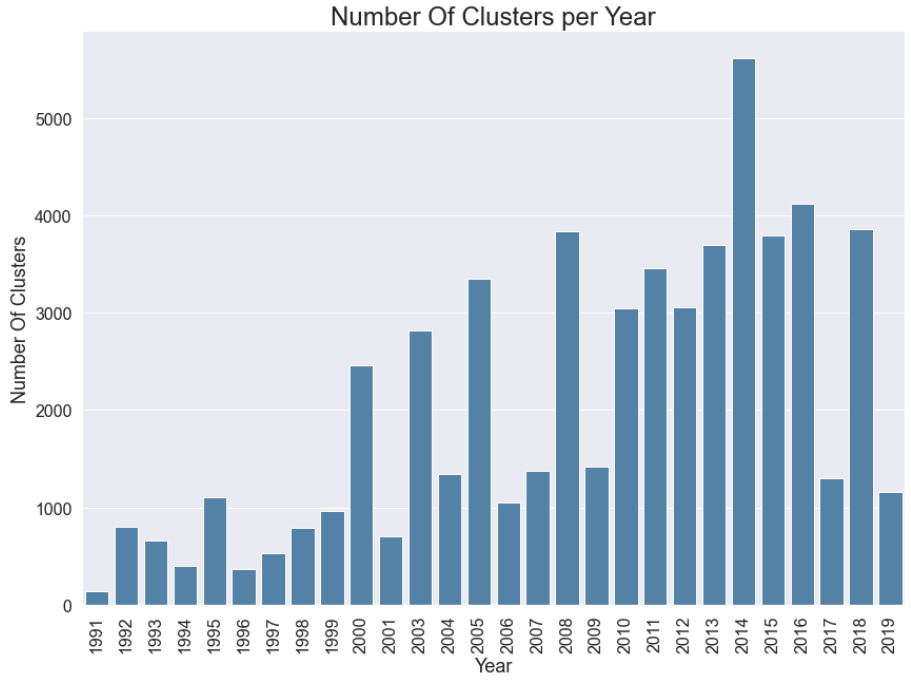


Figure 5.5: The figure represents the number of clusters per year. It can be seen that some years has been far more surveyed than other years. Both during the year of 2002 and 1990, no surveys has been done.

A closer look at the wealth asset distribution in Figure 5.6 shows also that South Africa and Morocco belongs to the wealthier countries in Africa. For some countries

great outliers can be seen with comparably low average wealth asset distribution such as Burundi, Malawi and Rwanda.

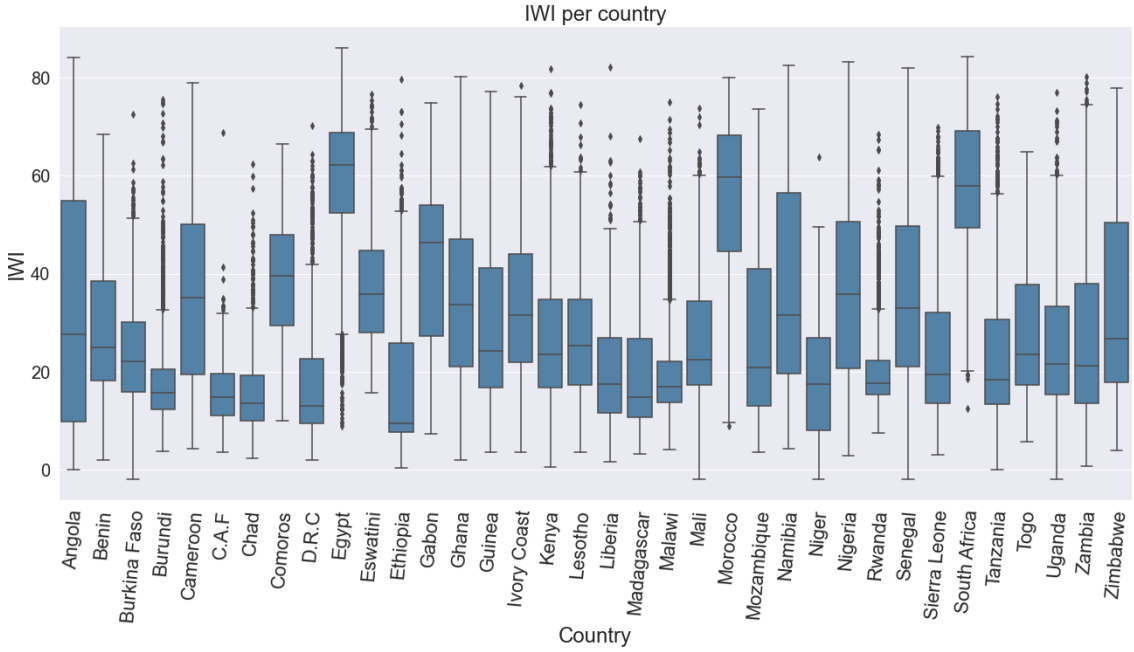


Figure 5.6: Wealth asset index distribution for each country. It proves easy to see which countries shares the larger wealth per household compared to other. Another interesting take-away is that even exceptionally poorer countries still has great outliers with some wealthier households.

5.5 Model evaluation

The models are evaluated using five fold cross validation, which essentially means splitting the dataset into five equally sized sections, called folds, to then train and evaluate the model on. Three of the folds are used as training data, for the model to tune its parameters to. One fold is used as validation data to evaluate the model's performance on unseen data during training. Finally, the last out of the five folds is used as a held out test set, to evaluate the performance of the final trained model which had the best performance on the validation set, see Figure 5.7. This process is repeated five times such that every fold is tested upon, and the performance is then averaged across all five test folds.

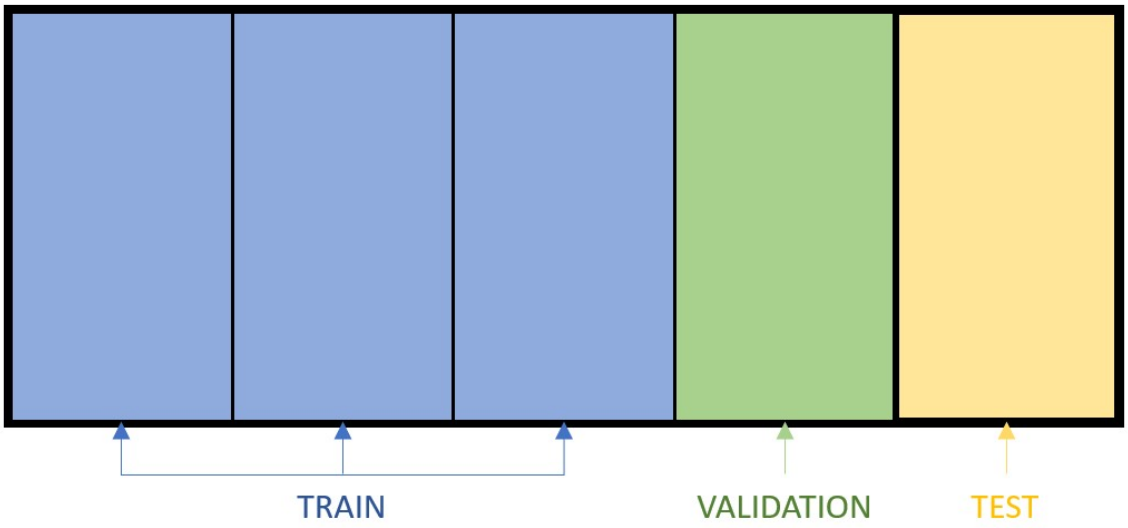


Figure 5.7: Cross validation. Data set divided into five folds, where three are used for training, one for validation, and one for testing.

One issue with setting up these folds is that the images are geographic data, meaning they are correlated by location. DHS clusters may lie in such close proximity to each other that an image of a cluster may contain parts of the image of another cluster. This means that denser areas are prone to contain large collections of overlapping images, see Figure 5.8 for an example. As can be seen in the right hand side of Figure 5.8, several images have large overlapping sections, meaning the same data is present in various different images. Naturally, these overlapping images need to be included in the same cross validation fold, so that no data from the training set also happens to be included in the test set.

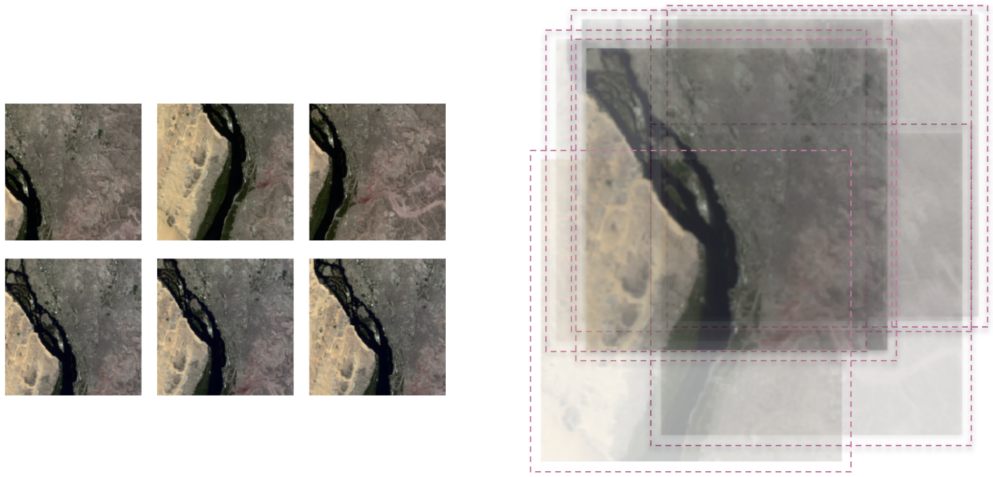


Figure 5.8: Visual representation of images of survey locations in close proximity overlapping each other to create large collections. The left hand side depicts six images of separate survey locations, and the right hand side depicts the images overlaid on top of each other.

Inspired by Petterson et al. [10], the approach taken to group the images into folds was to use the clustering algorithm DBSCAN to first create clusters of overlapping images, to then assign the clusters to the folds. This ensures that no overlapping images are assigned to different folds. DBSCAN works, as more thoroughly described in Chapter 3.5, by assigning points within a certain distance of each other to the same cluster. The distance where images might overlap is not entirely trivial to calculate, as the dimensions of an image are given in meters, whereas the location of the image is specified in degrees longitude and latitude. Since the Earth is roughly a sphere, one degree longitude or latitude equates to different distances in meters depending on where the measurement is taken. For example, traveling 360 degrees around the world directly east starting from South Africa would result in a shorter distance traveled as opposed to traveling 360 degrees directly east along the equator. Therefore, in order to find a fitting value for the distance parameter in DBSCAN, the maximum possible longitude and latitude spanned by an image was calculated. One degree latitude or longitude spans the longest distance along the equator and prime meridian (i.e. 0 degrees longitude and latitude), where one degree equates to 111 km, which is $\frac{1}{111}$ degrees/km. The input images are 224 x 224 pixels with a resolution of 30m per pixel, which corresponds to a side length of $224\text{px} \cdot 30\text{m} = 6.72$ km. This all amounts to a maximum side length of $6.72\text{km} \cdot \frac{1}{111}\text{degrees/km} \approx 0.0856..$ degrees. The distance parameter of DBSCAN is therefore set to 0.0856, and the minimum amount of points to create a cluster is set to 2, such that an image only needs to overlap with one other image to create a cluster.

Three separate sets of cross validation folds are constructed. One set of folds, presented in Chapter 5.5.1, is assigned such that all data within a country is assigned to the same fold, essentially meaning that the folds are comprised of disjoint sets of countries. These folds are called *country folds* since they contain entire countries. The second set of folds is assigned without this restriction, such that points within a country can be assigned to different folds. These folds are called *in-country folds*, since data can be split into separate folds *in countries*. The third set of folds are called *year folds* and are constructed by grouping sets of surveys based on the year they were conducted in. The reasoning for having three sets of cross validation folds is that the in-country folds test the general performance of the model on countries it has trained on, whereas the country folds and years folds test the model on countries and years in which it has not seen before. Given that the semi-supervised model has been pretrained on more data, which also spans more countries and regions, the hypothesis was that it should be more generalized, and therefore have better performance on extrapolating to countries and years which it has not had supervised training on compared to the entirely supervised baseline model. Further, since we will perform both spatial and temporal self-supervised pretraining, it is reasonable to evaluate the performance of the model in a temporal setting with the years folds, and a spatial setting with the country folds.

First, the general process of assigning the clusters or countries to folds is presented along with the creation of the in-country folds, and then the construction of the country folds is described.

5.5.1 In-country folds

When distributing countries into folds, we want the folds to have somewhat similar distributions. This means that the folds are stratified such that each fold should have similar distributions of the target variable, the IWI, and contain roughly the same amount of images from any given country.

The clusters created by the DBSCAN algorithm are assigned to five cross validation folds according to a process aimed at creating folds with similar IWI distributions. The general process is to assign clusters to folds, one at a time, by finding the most "suitable", non-full fold to assign the current cluster to. Since the five folds should be of similar sizes, the folds are limited to hold 20% of the data. If assigning a cluster to a fold will make the fold overfull, the cluster is assigned to the next most suitable fold which will not be overfilled. The clusters are sorted by size in descending order, such that the largest cluster is assigned first and the smallest cluster is assigned last. The reasoning for this is that assigning the largest clusters first allows one to have more granular control of the fold sizes since one can fill out the remaining space in a fold with smaller clusters, ensuring that the fold sizes are evenly distributed.

Now, how suitable it is to assign a cluster to a fold can be based on a multitude of heuristics. One example could be to always assign the current cluster to the fold which currently has the least amount of data in it, which was done by Pettersson et al. [10]. However, this approach doesn't provide any measures to ensure that the folds are fairly distributed with respect to the target variable, the IWI. This means that one fold could, in the worst case, contain the majority of the locations with high IWI, and another fold could contain the majority of locations with low IWI. Consequently, an imbalance would be imparted upon the dataset, where the model would be trained on one distribution, to then be evaluated on an entirely separate distribution.

In order to mitigate this imbalance, the *suitableness score* used in this project is instead based on moving the mean of a fold closer to the actual mean of all of the data. More specifically, if a cluster were assigned to a fold, the closer the mean of the fold is to the actual mean of all data, the higher the suitableness. This consideration makes the distribution within folds more likely to resemble the distribution of all of the combined data, at least in terms of the mean. The suitableness score also includes a penalty for assigning a cluster to a fold that is larger than the rest of the folds. The full equation can be seen in Equation 5.1, where n_{fold} is the number of folds currently in the fold, λ is the *big fold penalty*, μ_{fold} is the mean IWI of the fold if the current cluster was assigned to it, and μ_{total} is the mean IWI for all data.

$$\text{suitableness} = \frac{n_{fold}}{\lambda} + |\mu_{fold} - \mu_{total}| \tag{5.1}$$

The big fold penalty is added as a measure of dispersing the larger clusters across folds instead of placing them into the same fold. Since the penalty is relative to

the size of the fold, it is most apparent when the relative differences in fold sizes, i.e. the quotient of fold sizes, are large. This is the case at the beginning of the process, as assigning a large cluster to a fold of small size means a higher relative increase in size. Thus, large clusters are more likely to be distributed in separate folds than without including a big fold penalty. This is desirable since having folds of evenly distributed sizes along the assigning process means fewer folds will quickly become overfull, allowing more options of folds for each cluster to be assigned to. The importance of the big fold penalty then diminishes along the process as the fold sizes become closer on a relative scale, which allows the process to put more weight on creating even IWI distributions among folds. It also provides a tuneable parameter that can be adjusted to alter the distribution of clusters among folds, allowing one to try multiple values to then select the one resulting in the most evenly distributed folds.

The folds are depicted on a map of the African continent in Figure 5.9, and the IWI distribution of folds can be seen in Figure 5.10.

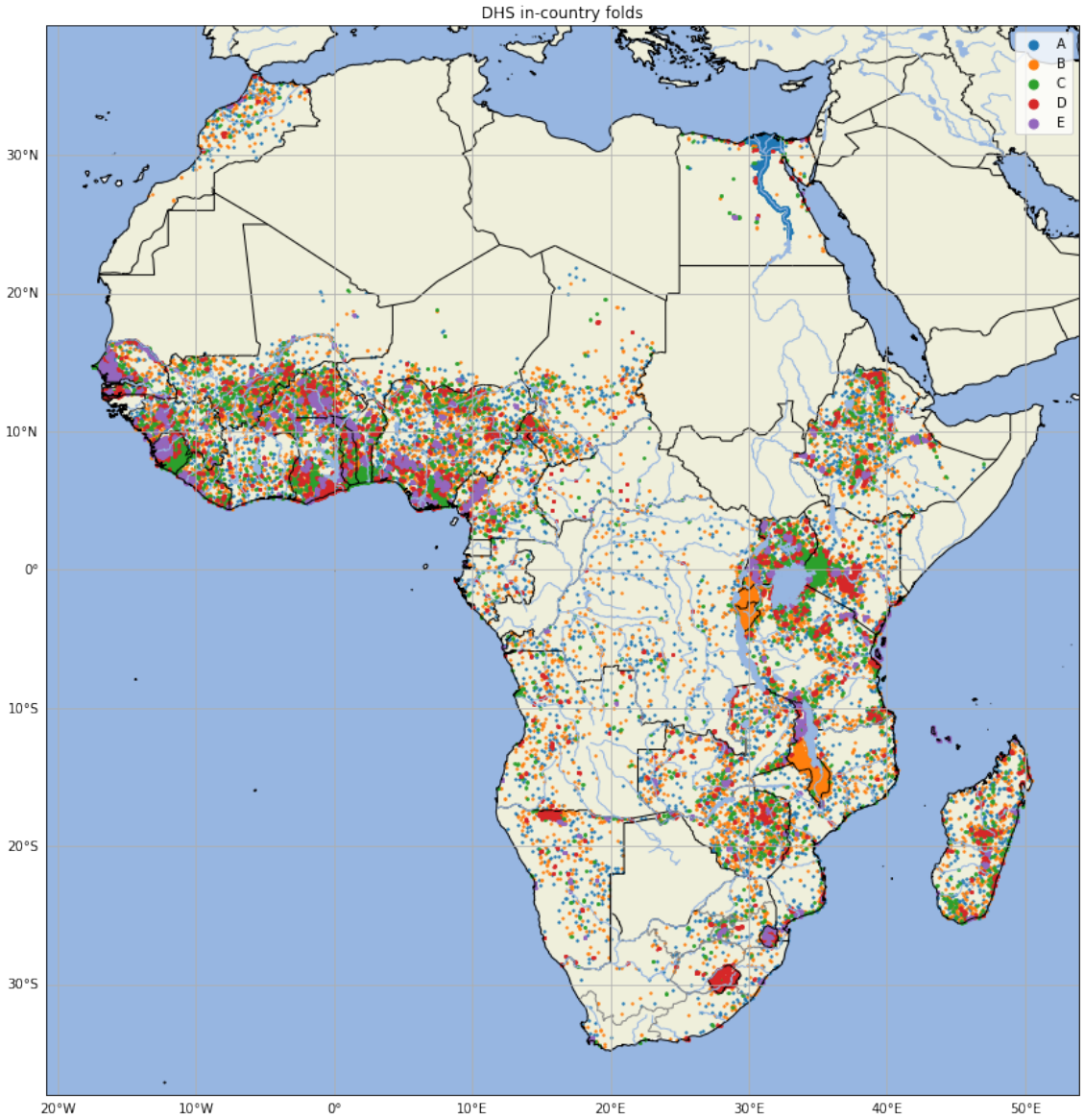


Figure 5.9: Initial cross validation folds composed from clusters generated by DBSCAN.

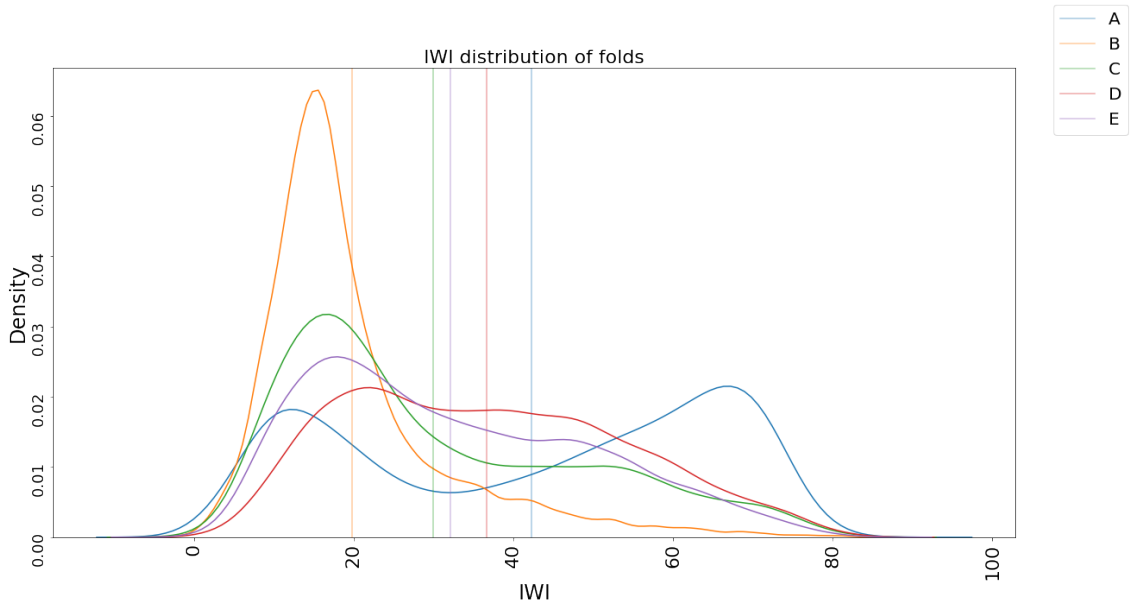


Figure 5.10: Distribution of IWI across folds. The vertical lines depict the mean of the fold with the corresponding color. Notice that the distributions in IWI differ wildly.

In Figure 5.10 we can see that the data has clearly not been separated into folds with equal IWI distribution, as the distribution curves look very dissimilar between folds. In Figure 5.9 we can see that some data points, e.g. along the Nile, are so densely distributed that they are all included in the same cluster, and therefore also in the same fold. As can be seen in Figure 5.11, the sizes of clusters differ wildly, and specifically, the cluster along the Nile contains over 6000 data points as can be seen in Figure 5.12. Since the IWI-distribution is not consistent across the entire continent, it is important that not all data points within a country are assigned to the same fold. This is especially important for Egypt, as it contains the most amount of data points, and also has the highest mean IWI out of all countries. This is the reason that the folds show different distributions. As can be seen on the map in Figure 5.9, the cluster along the Nile is included in fold A (blue), which explains the bump at 70 IWI in the distribution of fold A in Figure 5.10.

To combat the issue of large clusters skewing the distribution of IWI, some data points were removed from the dataset to break the chains of overlapping images into separate clusters. This was done by first visually inspecting the largest clusters, which can be seen in Figure 5.12, to then remove points within a less dense region of a cluster in order to break it apart. Removing mainly images of sparsely populated areas might however induce some bias in the dataset, since rural areas are generally less wealthy than urban areas. Still, this compromise was deemed necessary since one would have to remove far more urban data points in order to break the chain in a cluster, meaning greater loss of data. Through trial and error, more and more regions of data were removed to break the large clusters apart, and in total, 417 points were removed, corresponding to roughly 0.73% of the entire dataset, a sacrifice deemed

worthy in order to create fair folds for training and evaluating the model.

The process of clustering overlapping images and assigning the clusters to folds according to their suitableness score was then repeated without the removed points. The final folds are visualized in Figure 5.13, where one can see that e.g. the cluster along the Nile has detached into separate clusters, and separated into different folds. Consequently, the similarity in distribution of IWI across folds has greatly increased, as can be seen in Figure 5.14 and Table 5.3.

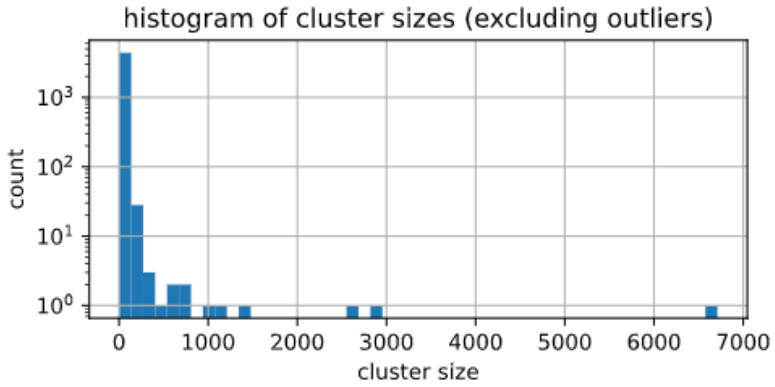


Figure 5.11: Histogram of sizes of clusters generated by DBSCAN.

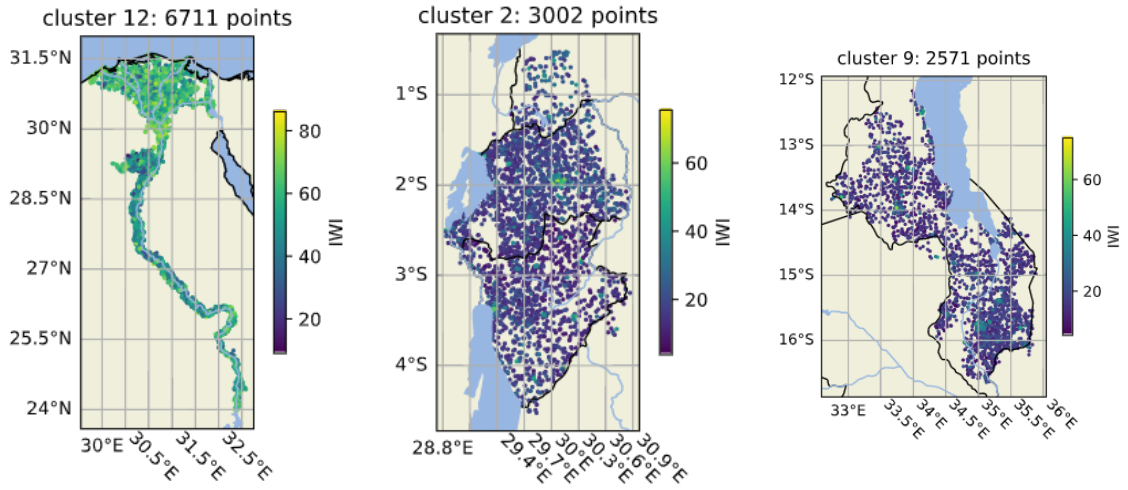


Figure 5.12: The three largest clusters generated by DBSCAN. Note that the leftmost cluster along the Nile not only contains more than 10% of the dataset, but also has generally very high IWI values.

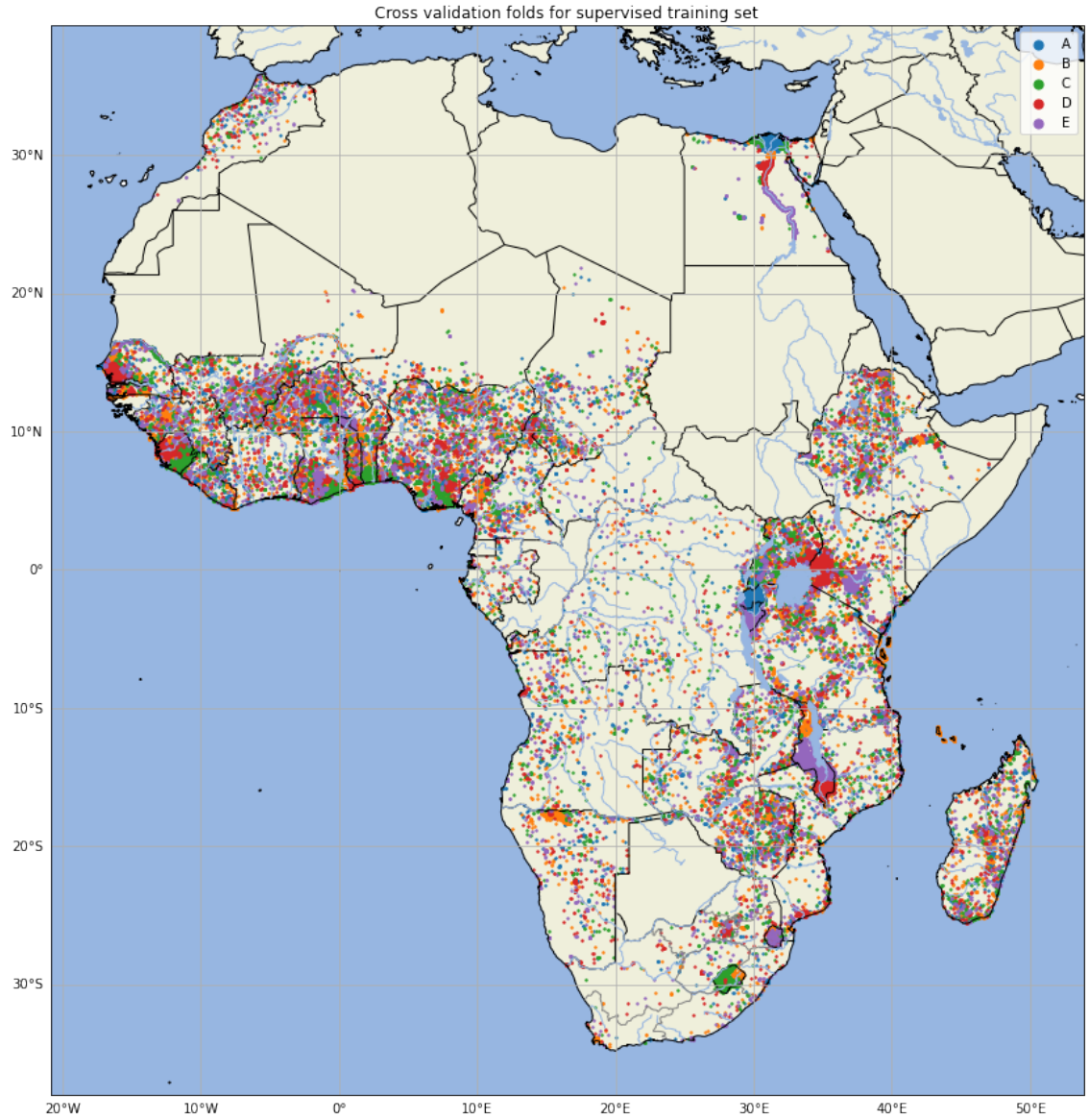


Figure 5.13: Cross validation folds where long chains of overlapping images have been separated. Notice that e.g. the cluster along the Nile has been detached into separate clusters and folds.

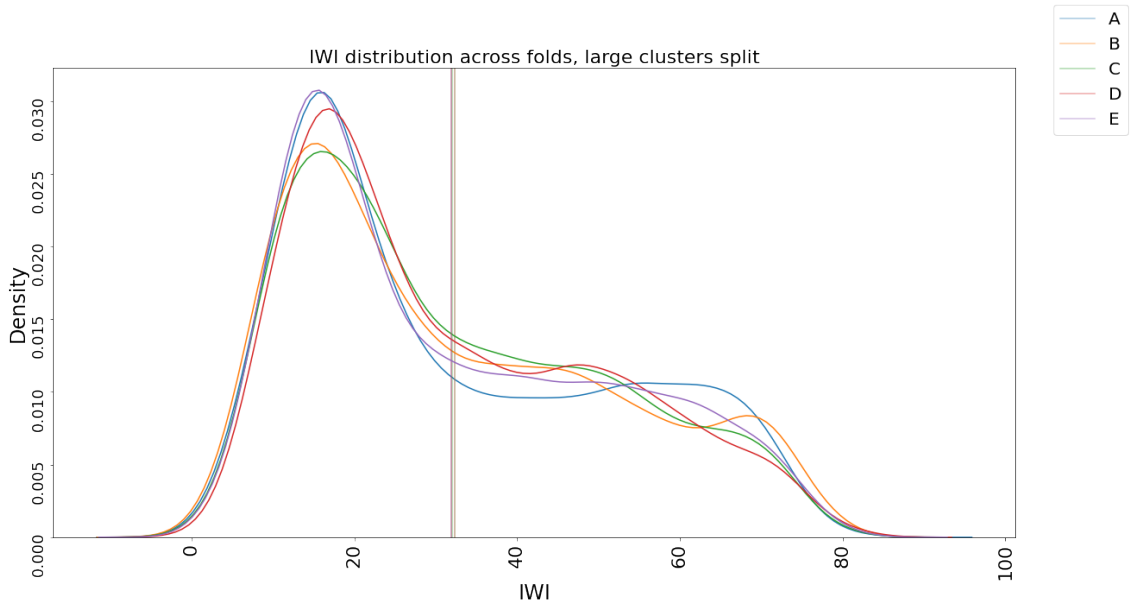


Figure 5.14: Distribution of IWI for cross validation folds. The vertical lines depict the mean of the fold with the corresponding color.

Fold	Mean	Standard deviation	Size
A	32.30	19.99	11353
B	32.29	19.91	11355
C	31.98	18.88	11356
D	31.96	18.71	11356
E	31.94	19.50	11356

Table 5.3: Fold statistics for in-country folds.

5.5.2 Country folds

The country folds are created in a similar manner as the in-country folds, with the exception being that every data point within a country must be included in the same fold. First, the data is grouped by country, and then each country is assigned to a fold, one by one, according to the same process of assigning the in-country folds. I.e., the countries are sorted by size, then the suitableness of assigning the country to every fold is calculated, and the country is then assigned to the most suitable fold which will not be overfilled. The countries included in the different folds can be seen in Table 5.4, and are also visualized on the African continent in Figure 5.15. The distribution of IWI across folds can be seen in Figure 5.16. Note that the distributions are not as similar to each other as the distributions of the in-country folds are. This is due to the fact that grouping the data by countries instead of grouping by clusters created by DBSCAN, makes each set of data points to be assigned to a fold at any given time larger, which grants less granular control of the average IWI of any given fold. Specifically, one can note that folds B to E have roughly similar distributions and means, while fold A (blue) has a drastically

different distribution. This is explained by the fact that fold A contains Egypt, which is the country having both the most amount of surveyed clusters, and also has the highest mean IWI, both by a large margin (see Figures 5.4 and 5.6).

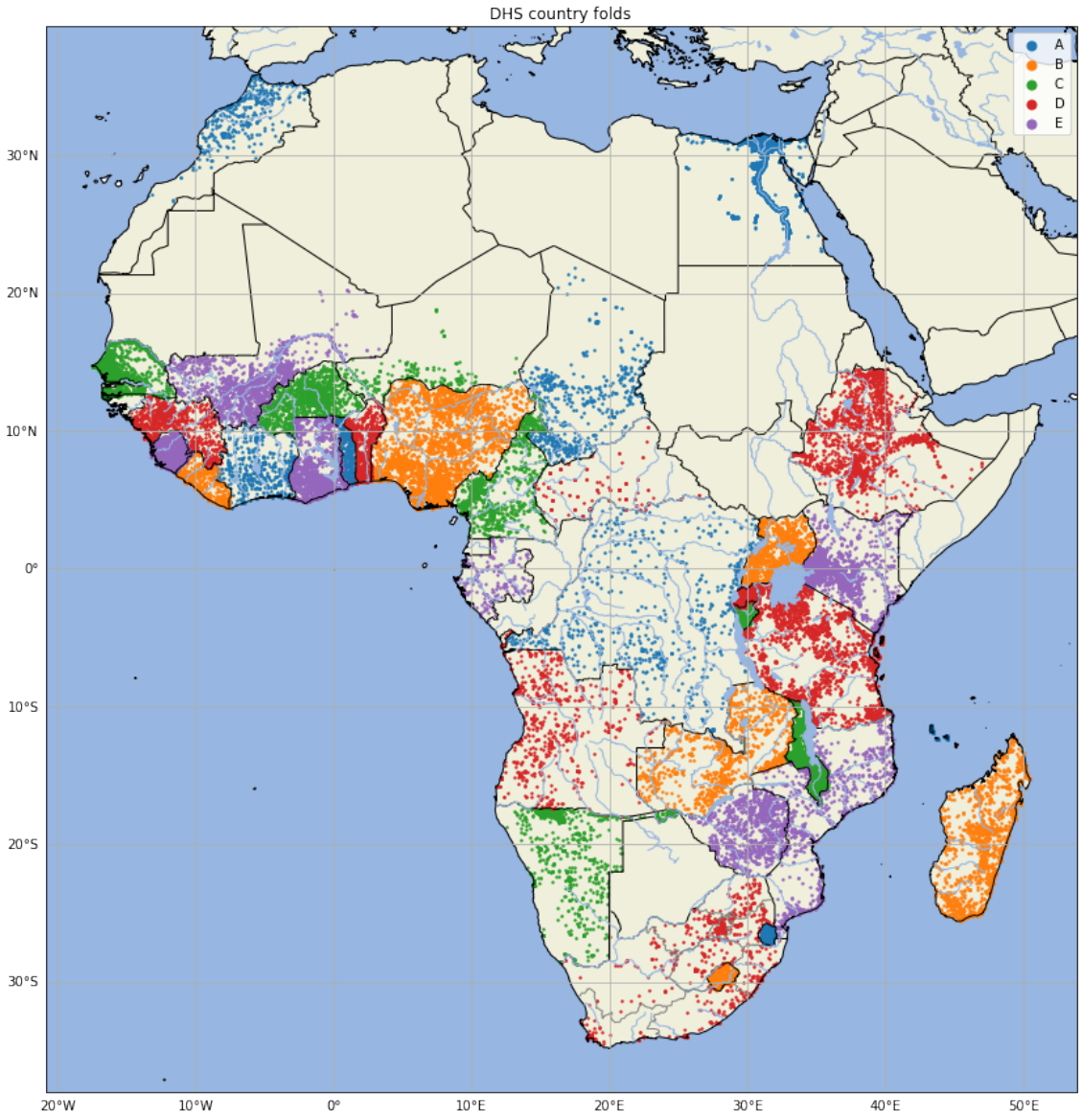


Figure 5.15: Country cross validation folds

Fold	Countries
A	Egypt, Democratic Republic of Congo, Togo, Ivory Coast, Chad, Morocco, Eswatini, Comoros
B	Nigeria, Uganda, Madagascar, Zambia, Lesotho, Liberia
C	Malawi, Senegal, Burkina Faso, Cameroon, Namibia, Burundi, Niger
D	Tanzania, Ethiopia, Benin, Rwanda, Guinea, Angola, South Africa, Central African Republic
E	Kenya, Ghana, Mali, Sierra Leone, Zimbabwe, Mozambique, Gabon

Table 5.4: Countries included in each fold

Fold	Mean	Standard deviation	Size
A	49.33	20.20	11260
B	28.49	16.73	11271
C	27.56	16.21	11348
D	26.07	17.56	11475
E	29.23	15.88	11422

Table 5.5: Fold statistics for country cross validation folds

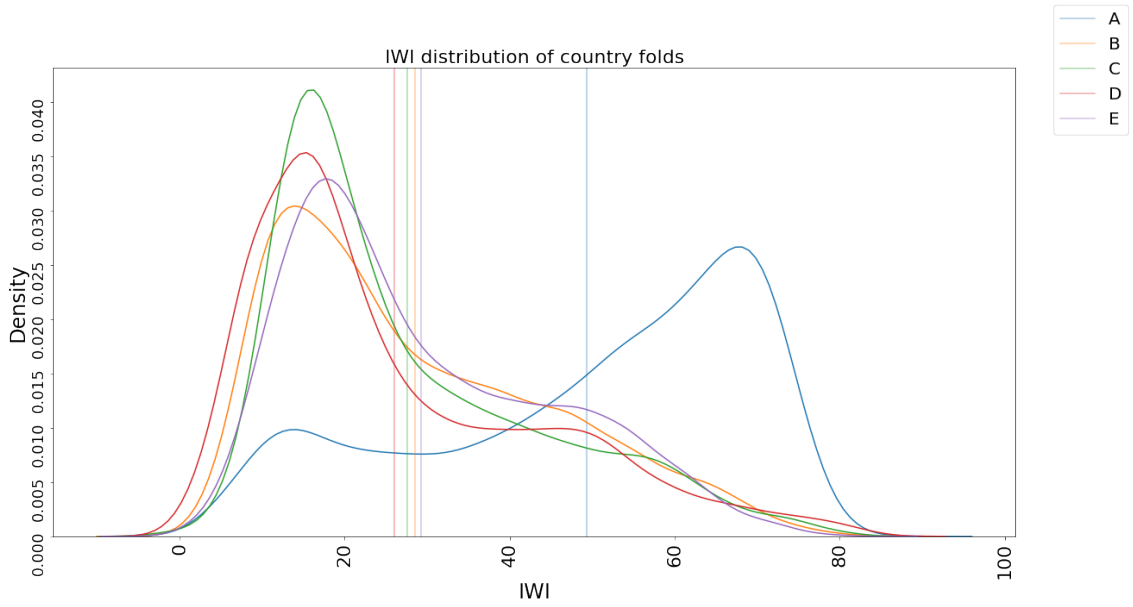


Figure 5.16: Distribution of IWI for country cross validation folds. The vertical lines depict the mean of the fold with the corresponding color.

The country folds are not used in the exact same manner as the in-country folds when performing cross validation. The country folds depicted above are the **test** folds used for evaluating the model. I.e. for each test fold, the model is trained on the rest of the data and then evaluated on the test fold. When using the in-country folds, one of the remaining four folds was used for validation, and the other three were used for training. Ideally, one wants the training and validation set to

be similar to each other, and this was the case for the in-country folds. However, since the country folds are split by country, using one of the folds as the validation set will mean the validation set only contains images of countries not present in the training set. This means that the best performing model chosen to be evaluated on the test set of countries it has not seen would be chosen by its performance on another, different, set of countries it has not seen. Therefore, the split between the validation and training set from the remainder of the data (without the test set) disregards any country borders.

Given that 20% of the data has been held out for testing, four folds are created from the remaining 80%, where one fold should be used for validation, and the remaining three for training. These four folds are created in the same way the in-country folds are created, by using DBSCAN to first cluster overlapping images, and then assigning them to the most suitable fold. Then, one of the folds is chosen for being used as the validation set. All the folds are relatively similar in IWI distribution, so any of the folds could essentially be selected as the validation set, but here the method is to choose the fold that has the mean closest to the overall mean of all the four folds combined. A visualization of the test, train, and validation sets for test fold A can be seen in Figure 5.17. Note that the train and validation data are interspersed with each other, while the test data lies in countries entirely disjoint from the train and validation sets.

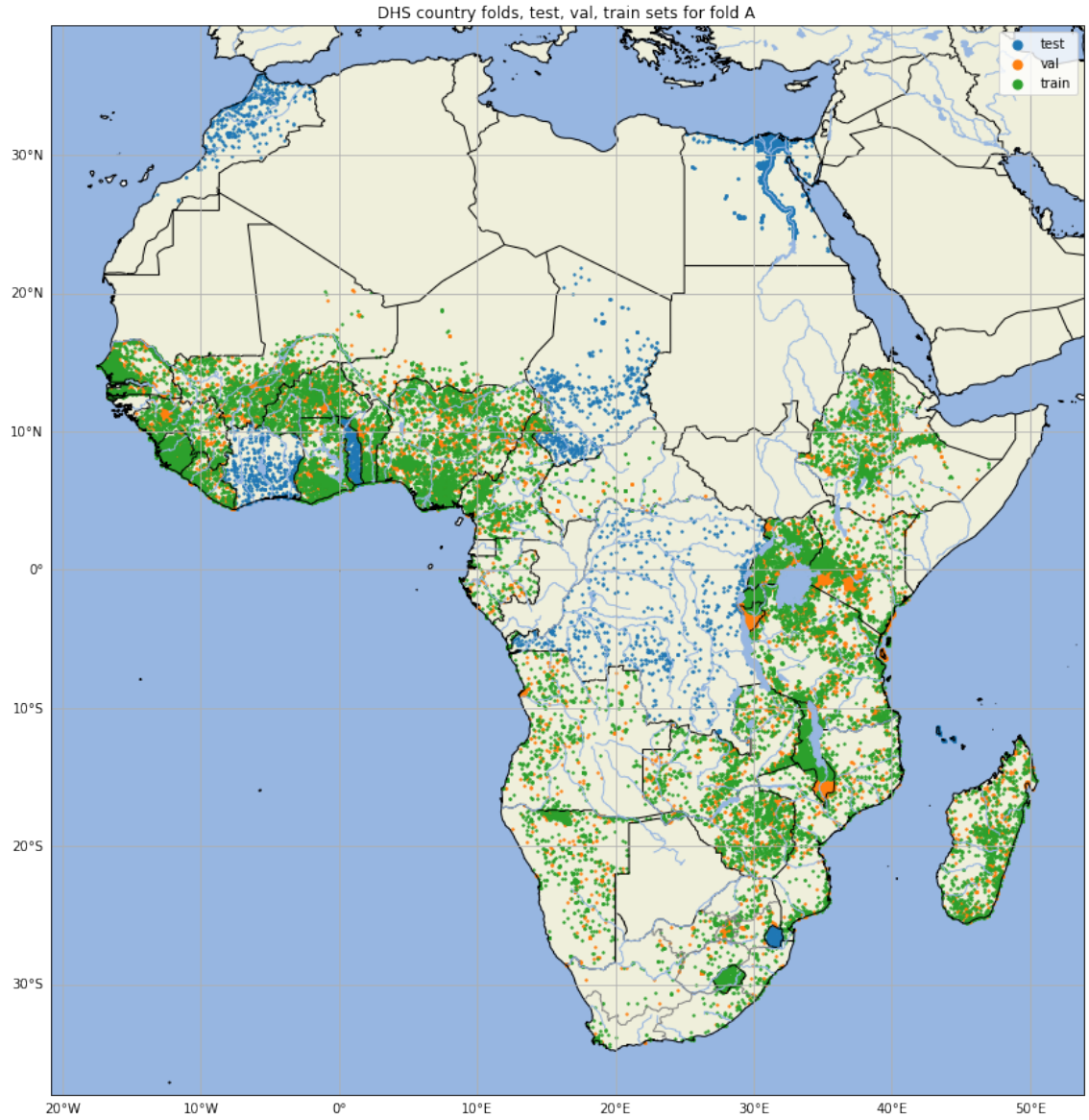


Figure 5.17: Visualization of which surveys belong to the test, train and validation sets for fold A. Test data lies entirely in countries disjoint from the train and validation sets.

5.5.3 Year folds

In order to evaluate the models ability to extrapolate temporally, i.e. the performance on years it has not trained on, the model is also tested on held-out years. The data set contains 10 three-year timespans in total, starting with 1990-1992 and ending with 2017-2019, see Table 5.6. To create the year folds for cross validation, sets of two consecutive timespans are combined such that the first fold contains 1990-1995, and the last fold contains 2014-2019, see Table 5.7. The train and validation sets are then constructed from the rest of the data with the same approach as for the country folds, such that the train and validation set contain all years and

’90-’92	’93-’95	’96-’98	’99-’01	’02-’04	’05-’07	’08-’10	’11-’13	’14-’16	’17-’19

Table 5.6: Timespans included in the data set.

Fold years	**Fold size**
’90-’95	3050
’96-’01	5764
’02-’07	9846
’08-’13	18396
’14-’19	19720

Table 5.7: Folds created for held out years evaluation.

have roughly similar IWI distributions. The model is than trained on all years but the timespan of the test set, and evaluated on the test set.

6

Model implementations

This chapter describes the methods used to develop the machine learning models. Firstly, both the self-supervised pretext tasks and the supervised downstream task are defined. What follows is a description of the proposed model architectures and model implementations. That includes every proposed pretext task and the downstream task. Additionally, the ways of evaluating the results brought forward by the downstream task is covered.

6.1 Self-supervised Learning

In the following sections, the various pretext tasks will be presented. These will be presented with emphasis on their temporal and spatial aspects. Three methods are proposed in this project, and all use contrastive learning to learn representations. The first one, *the temporal pairs model*, utilizes images that are from the same location but at different points in time. The other one, *the spatial pairs model*, instead utilizes images that are close in location but at the same time instance. The third one is the combination of the both, a *spatio-temporal model*, that takes images both in different instances of time and location.

6.1.1 Pretext tasks

In order to fully comprehend the benefits of applying self-supervision to a poverty prediction downstream task, one has to divide the problem. Since the task is spatiotemporal, meaning images differs both in time and in space, one can assume that different model settings can affect differently. Consequently, the self-supervised pretext task was trained with temporal and spatial augmentations separately, but also in combination. This results in a spectra of different settings for the contrastive pretext task. The structure for each of these pretext tasks is presented in Table 6.1.

Model	Temporal	Spatial	Regular augmentations	ImageNet pre-trained
Aug	No	No	Yes	Yes
Temporal	Yes	No	No	Yes
Temporal Aug	Yes	No	Yes	Yes
Temporal Aug (no ImageNet)	Yes	No	Yes	No
Spatial	No	Yes	No	Yes
Spatial Aug	No	Yes	Yes	Yes
Spatiotemporal	Yes	Yes	No	Yes
Spatiotemporal Aug	Yes	Yes	Yes	Yes

Table 6.1: Variations of proposed pre-text tasks.

What actually makes the model have a temporal or spatial aspect lies fundamentally in the model input. A purely temporal model will only compare images at different times, i.e. through contrastive learning. On the other hand, a spatial model does not incorporate time as a variable and every image could just as well be captured at the exact same time. In every aspect, augmentations can still be applied as well as having pre-trained ImageNet weights. Ayush et al. [6] initialized their models without weights pretrained on ImageNet, and therefore one model was also initialized in this way to determine whether starting with the weights from ImageNet increases performance. These adjustments could instead tell the story of whether the model is performing better, i.e. predicting poverty, based on which augmentations are used or if the network is pre-trained on ImageNet.

6.2 Augmentations

MoCo, and contrastive learning in general, relies on the concept of augmentations. Augmentations are used to create two slightly different views of an image, which are then passed through MoCo as a positive pair. As mentioned previously, MoCo aims to find similar feature representations for the images in a positive pair. In order for these feature representations to be meaningful and informative in some downstream task, the augmentations need to alter the image enough for it to not be trivial for the model to recognize if they are similar, but also not alter the image to the point where the similarities are lost.

The augmentations commonly used in the literature include rotating, zooming, blurring, gray-scaling and other image altering techniques. These will henceforth be referred to as regular augmentations. This project, inspired in part by Ayush et al. [6], takes the concept of augmentations one step further, to also include *temporal* and *spatial* augmentations, by augmenting the image within the dimensions of time and space. Since these different kinds of augmentations affect the image in different dimensions, they can not only be used on their own but also combined. In this section, general image augmentations are first described, then both temporal and spatial augmentations are covered.

6.2.1 Regular augmentations

The general image augmentations used in this project are based on the default image augmentations in MoCo. However, MoCo generally operates solely on RGB data, and therefore has some augmentations which are not trivially translated to multispectral data, such as grayscaling and color jittering. Due to time constraints, these specific augmentations have therefore not been implemented. This leaves three primary augmentations, flipping, blurring, and a random resized crop. In the augmentation pipeline, one image is first selected, and then a copy is made from that image. The copies are then augmented independently of each other. An image is always cropped to a square of random size between 0.08 and 1.0 times the side length of the original image and then rescaled to the original size of $224 \cdot 224$ pixels. Following this, a Gaussian blur, a vertical flip, and a horizontal flip are each independently applied with a 50% probability to the image. Some examples of augmented pairs can be seen along with the original image in Figure 6.1.

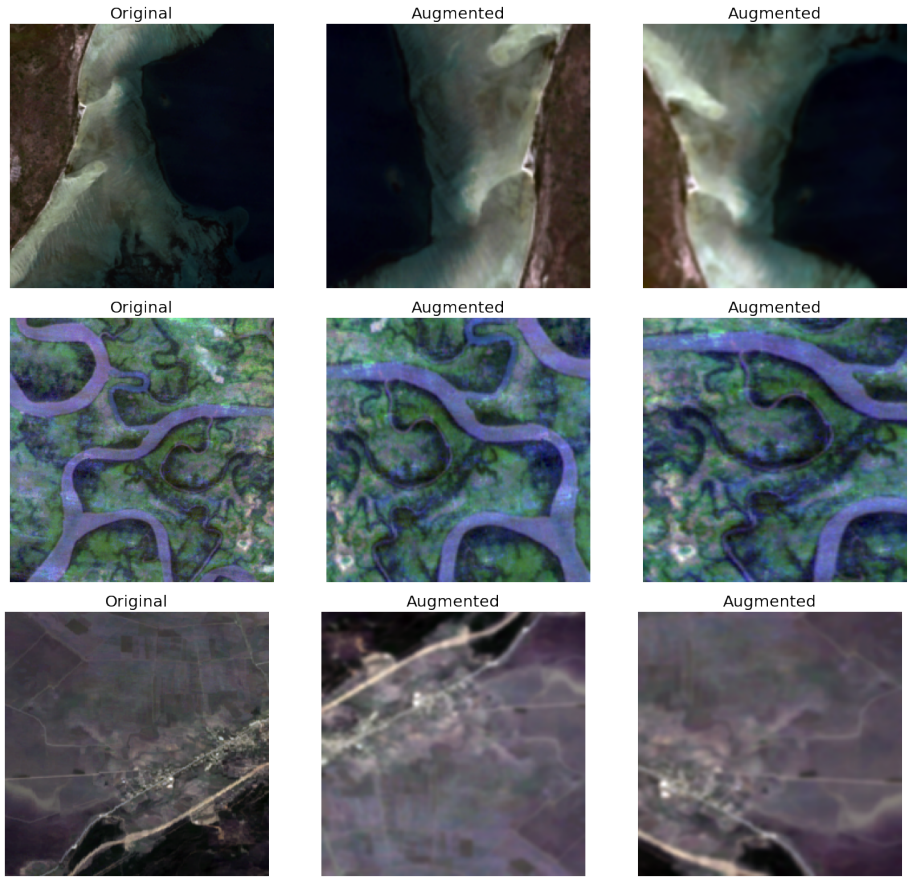


Figure 6.1: Examples of temporal pairs and augmentations for three locations. The left column contains the image of the location with no augmentations applied, and the middle and right columns depict a set of temporal pairs augmented independently.

6.2.2 Temporal augmentation

The next type of augmentation is a temporal augmentation. A temporal augmentation is simply selecting an image of the same location, but at a different point in time. Some examples of these temporally augmented images can be seen in Figure 6.2. Note that the locations can change quite drastically between timespans. For example, in the bottom row, except for the change in hue, a body of water seems to present in 2017-2019, while having a vaguely distinguishable contour in 2002-2004, and not being visible at all in 1996-1998.

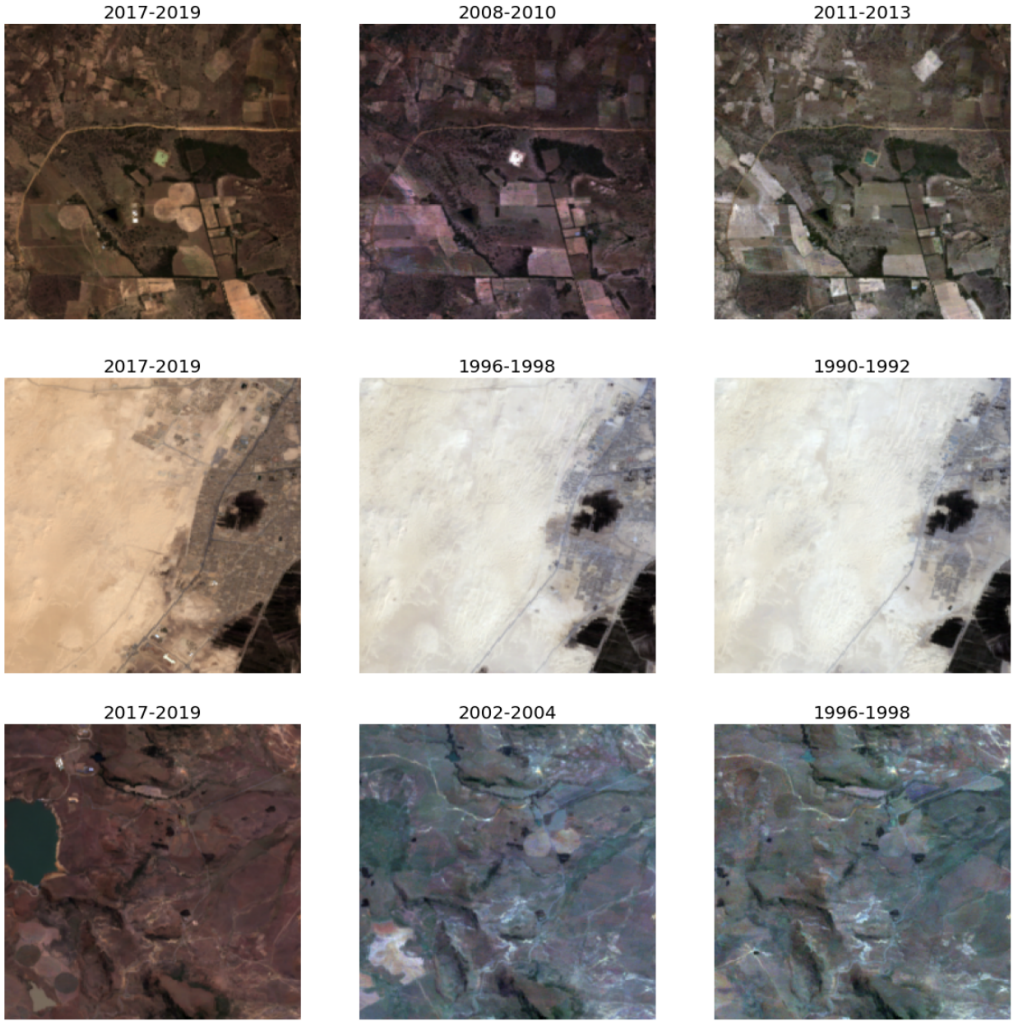


Figure 6.2: Examples of spatially aligned images from different time spans, i.e. temporally augmented images. The latest available image is shown in the left column, along with images of the same location from two other randomly selected timespans.

The data includes 10 three-year timespans in total, with the first being 1990-1992, and the last being 2017-2019. When generating a pair of temporally augmented images, first a random timespan is selected for the first image. The second image is then selected entirely randomly as well, with no regard to the first image except

for the fact that it cannot be the same timespan. This means that the positive pair formed can be any combination of timespans. Since the aim of MoCo is to extract similar feature representations from positive pairs, this should hopefully force the model to learn weights for extracting features that are generalized over the temporal dimension, meaning it is time-invariant. With this in mind, the temporal model applied in the downstream task should hopefully be more proficient at predicting poverty for timespans it has not trained on.

6.2.3 Spatial augmentation

Other than the temporal and regular image augmentations from MoCo, a *spatial augmentation* has also been implemented. In the same vein as considering selecting an image of the same place but at a different time as a temporal augmentation, we can consider selecting an image in the same timeframe but in close geographic proximity as a spatial augmentation. Figure 6.3, depicts some examples of these spatially augmented images which we will call *neighbor pairs*. As can be seen, the neighbor pairs bear a close resemblance to each other, and e.g. in the top right pair, one can see that the geographic location illustrated in the image to the right lies directly east of the location in the left picture. I.e., the image has been spatially augmented by moving to a nearby location.

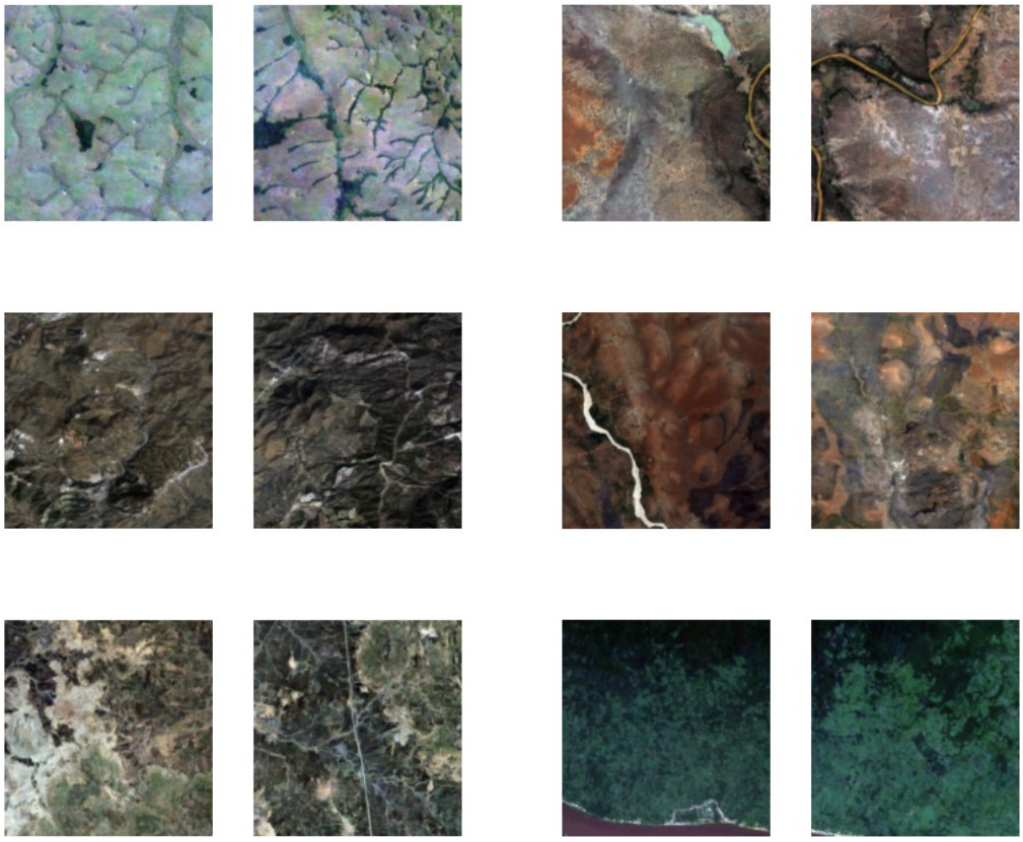


Figure 6.3: Examples of spatially augmented images - neighbor pairs.

Given that the neighbor pairs look similar, the augmented image still carries similar geographic features to the original image, however, not the exact same. This means that the model presumably needs to develop feature representations that are more general than those of the model training on images from the exact same location. This should also help the model generalize across regions that look similar but are not located close to each other.

As long as one does not go too far away from the original image to select a neighboring image, the pair should have similar geographic features. This radius could be varied and tuned in model training, but due to time- and computational constraints, only one single radius was evaluated. The options for radii can be seen in Table 6.2, along with the minimum, maximum, mean number of neighbors, as well as the number of images having no neighbors, for each proposed radius. Now, the broader the radius, the harder and more general the task is. One would like to select a radius which is large enough to have several options for selecting neighboring images, while not being too far away to be of an entirely different geographical region. As can be seen in Table 6.2, for all proposed radii there will always be points having no neighbors. In terms of the other statistics (min, max, mean), there are relatively large jumps between a radius of 5 and 6, 9 and 10, 13 and 14, and 14 and 15, and the values change relatively little outside of these jumps. Given this fact, there is a greater argument for selecting one of the radii immediately after a jump, i.e. 7, 10, 14, or 15, since we would like to keep the radius short while still giving several options for neighboring images. This led to the decision to select a radius of 10km, as it strikes a nice balance of short distance but with a lot of options for neighbors. As previously mentioned, one would ideally like to empirically evaluate more options to refine the selection to the radius which gives the best performance on the downstream task.

Another argument for selecting a radius of 10 km is that the locations of the DHS surveys for the supervised task are randomly displaced up to 2 km in urban areas, and up to 10 km in rural areas. This means that the maximum displacement of the survey location for any cluster is 10 km. Choosing a neighbor radius of 10 km will train the model to derive similar feature representations of any image that is within 10 km, and will therefore presumably generalize the model such that the impact of the displacement is lessened by the fact that the model would extract similar features within the entire possible displacement range.

Given the fact that the images have a side length of 6.72 km, and 124 645 locations out of 136 963 have no neighbors at a radius below 7 km as seen in Table 6.2, the overwhelming majority of neighboring images are between 6 and 10 km. In fact, a radius of 6.7 km gives 124304 points for having no neighbors, and a radius of 6.72 km gives 4607 points for having no neighbors. This means that barely any images contain overlapping locations with neighboring images, meaning the model cannot solely rely on any parts of the images depicting the same location and will have to develop more general feature representations.

Radius (km)	No N*	Min N*	Max N*	Mean N*
5	125475	0	4	0.13
6	124645	0	6	0.19
7	4607	0	11	3.26
8	4603	0	13	3.31
9	4602	0	17	3.37
10	3236	0	23	6.19
11	3234	0	25	6.27
12	3229	0	27	6.37
13	3228	0	31	6.47
14	2756	0	34	9.14
15	1834	0	45	14.22

Table 6.2: Table depicting min, max, mean, and amount of images having no neighbors for each radius. N* stands for neighbors. The selected radius of 10km is marked in yellow.

6.3 Downstream task

The downstream task is effectively the poverty prediction model. In this section, the baseline model is first presented which acts as a means of reference to the other models. The main function is to contextualize the results of the other models powered by self-supervision. Thereafter, the proposed semi-supervised contrastive models are presented and their corresponding experiment set-ups.

6.3.1 Baseline Model

The base neural network architecture used for the models is ResNet-18. The main reason for this is due to its relatively small size, meaning low training time compared to deeper models such as ResNet-50, and high performance on ImageNet classification, as well as its prominence in related works such as Yeh et al. [8].

The model is composed of two separate ResNet-18s, one for multispectral data, and one for nightlight data. The output of these is then concatenated and fed through a linear fully connected layer, similarly to Yeh et al. and Pettersson et al. [8, 10]. Both ResNets are pretrained on ImageNet, which is an extensive dataset used for classification, containing more than 14 million images from 1000 categories of various objects, such as car, centipede, screwdriver, and pomegranate [25]. This pretraining enables the model to attain weights useful for extracting general features of an image, such as lines and corners, which can serve as a better baseline for training on a specific task than just random initialization. Given that the features learned by the ResNet are very general, the pretraining will be useful even though the domain of ImageNet and satellite imagery are drastically different.

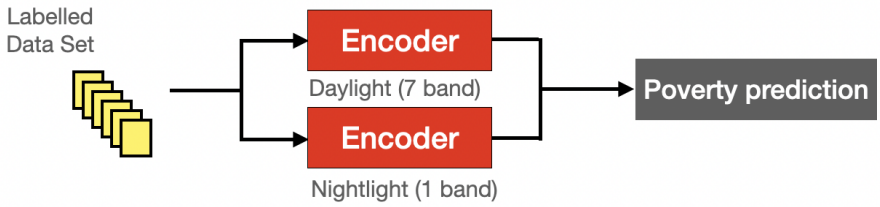


Figure 6.4: Baseline model architecture inspired by Yeh et al.[8].

A regular pretrained ResNet takes three channels as input: red, green, and blue, as all images in ImageNet are represented with this format. Our multispectral ResNet needs to take seven channels as input, and the nightlight ResNet only takes a single channel. The input layers of the ResNets are therefore altered to take this change into account. The weights for the red, green, and blue channels of the multispectral ResNet are simply copied from the pretrained ResNet, and the weights for the remaining channels are copied from an arbitrary channel, in this case, the blue channel. Other approaches, such as the one applied by Yeh et al., include initializing the multispectral layers with the mean of the red, green and blue channels of the pretrained weights. The final layer of the model is also adjusted from the layer of size 1000 used for ImageNet classification to have only a single output node to make it suitable for regression by outputting a single scalar.

6.3.2 Semi-supervised contrastive models

Here, the semi-supervised procedure of pretraining our model using self-supervised learning (SSL), to then fine-tune using supervised training is described. A general overview of the semi-supervised pipeline with a generic pre-text task can be seen in Figure 6.5. As can be seen, the encoder, depicted with a pink rectangle, is first trained on the pre-text task. The model is then copied and acts as a starting point for training on the supervised downstream task of predicting poverty, i.e. transferring the weights from the pretext task to the supervised ResNet18 encoder. All encoders are of type ResNet18 and the other steps follow from what is described in the theory section about the MoCo architecture, see 3.6.1.

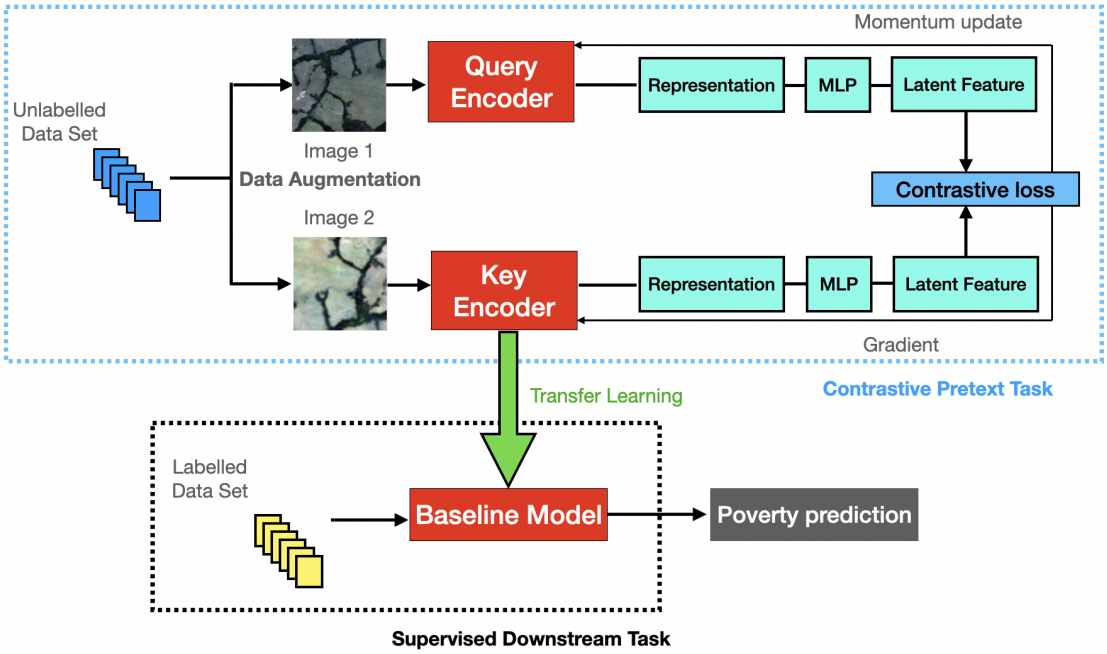


Figure 6.5: Pipeline showcasing the interaction between the contrastive pretext task based on temporal pairs and MoCo-V2 framework, with the supervised downstream task which was inspired by Yeh et al.[8].

The purpose of the pipeline is to deliver an overview of the main concepts involved with self-supervised learning. One attempts a certain pretext task with the use of an abundant data set whose knowledge is transferred to a supervised downstream task. The pretext tasks follow the set-ups earlier presented in Table 6.1, which basically tells what augmentations have been made, if the network is pre-trained on ImageNet and what aspect the model is focused on, i.e. spatial, temporal, or both.

6.4 Experiments set-up and post-model calibrations

In this thesis, a plethora of different models with different settings was implemented. Recall the variations of proposed-pretext tasks from section 6.1.1 pretext tasks, where all models stated have been implemented. In addition to these, a baseline model is used as a benchmark to evaluate whether SSL improves overall accuracy in poverty predictions. Additionally, the models are evaluated in three ways, one which employs the in-country folds approach, one that uses the country folds approach and one that uses the out-of-year folds. The main difference between the two is that country folds have the exception that every data point within a country must be included in the same fold. Shared between the three approaches of evaluation, they are trained using 5-folded cross-validation. This means that each model was trained 5 times, each time holding out a different fold from the data set for testing. The results were reported using the performance statistical metric r^2, to add rele-

vance when comparing to prior research [8, 9, 10]. A post-model calibration was also performed involving The International Wealth Index(IWI). The IWI values can only span from 0 to 100, hence a capping of the predicted values was made. This means that any predicted IWI values reaching higher than 100 were set to 100. In short, these models were implemented when addressing the question of whether SSL improves accuracy in poverty predictions:

- **Baseline model**: Same architecture as the model produced by Yeh et al., taking a single image as input [8].

- **Temporal MoCo**: A temporal pretext task that takes images at the same locations but at different points in time. Tested with and without regular augmentations. This model was also tested with and without pre-trained ImageNet weights.

- **Spatial MoCo**: A spatial pretext task that takes images in close spatial proximity. Tested as well with and without regular augmentations.

- **Spatiotemporal MoCo**: A spatiotemporal pretext task that takes images both in close spatial proximity as well as at different points in time. The same set-up of augmentations is used here also.

7

Results

The results chapter covers the performance of all proposed models including the baseline models. Additionally, recall the three evaluation methods created through cross-validation: The in-country folds, country folds, and out-of-year folds. The purpose is to find whether models based on spatial, temporal, or spatiotemporal aspects should perform better and be more generalizable. For every cross-validation set-up, a baseline model is created and compared to the proposed semi-supervised models.

7.1 Predictive performance of semi-supervised models

In the following section, an evaluation is made to decide if models aided by self-supervised learning improve the performance of the benchmark baseline model. The baseline models act as benchmark for this thesis results. The baseline models are inspired by prior research in poverty predictions by Yeh et al. [8]. The main objective is to determine whether self-supervised learning can improve these results. If that is the case, all further models would gain significant benefits from being pre-trained on remote sensing data compared to the ImageNet alternative.

Baseline model results	r^2
In-country	0.6366
Country	0.6185
Out-of-years	0.6460

Table 7.1: Baseline benchmark model r^2 across all evaluation methods: **In-country folds**, **country folds** and **out-of-year folds**.

In the following sections, the best performing results for each evaluation method are presented and these include the use of in-country folds, country folds, and out-of-year folds. The best performing proposed model is compared to its corresponding baseline model, and its performance per country is also presented in section 7.7. Additionally, an investigation is made on how much the prediction deviates, on average, from the actual values in the data set.

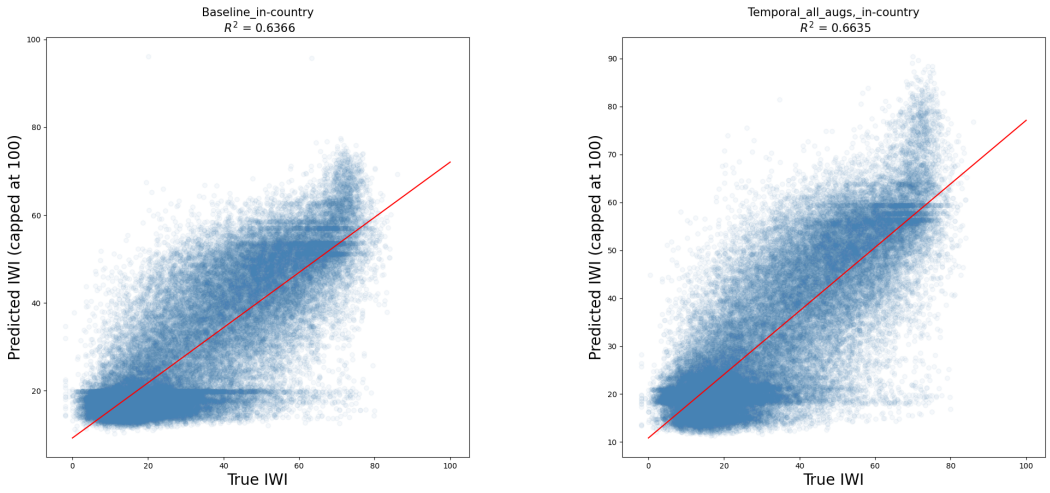
7.2 Performance with in-country folds evaluation

For the in-country folds, the r^2 values for each of the proposed models can be seen in Table 7.2. Recall that the in-country folds are intended as a general measure of the predictive power of the models. The only model which shows a significant leap in performance as well as it is surpassing the baseline model is the temporal model with all augmentations, marked in green in the table. The same model but without being initialized with ImageNet weights before SSL training, and the spatial model, both marked in yellow, also improves upon the baseline, but not to the same extent.

Model	r^2
Aug	0.6352
Temporal	0.6344
Temporal Aug	0.6635
Temporal Aug (no ImageNet)	0.6458
Spatial	0.6411
Spatial Aug	0.6143
Spatiotemporal	0.6241
Spatiotemporal Aug	0.6336

Table 7.2: Results on in-country folds in r^2. Aug denotes a model with all augmentations. Temporal/spatial/spatiotemporal decides the aspect of the model. One of the models is not initialized with weights pre-trained on ImageNet, hence the "no ImageNet" clarification.

For the best performing model shown in Figure 7.1 one can observe a ca. 3 percentage point increase in performance compared to the baseline benchmark model.



(a) **(b)**

Figure 7.1: Results from baseline model on the **a) in-country** folds and the best performing model **b)** which is a temporal model pre-trained on ImageNet with all augmentations.

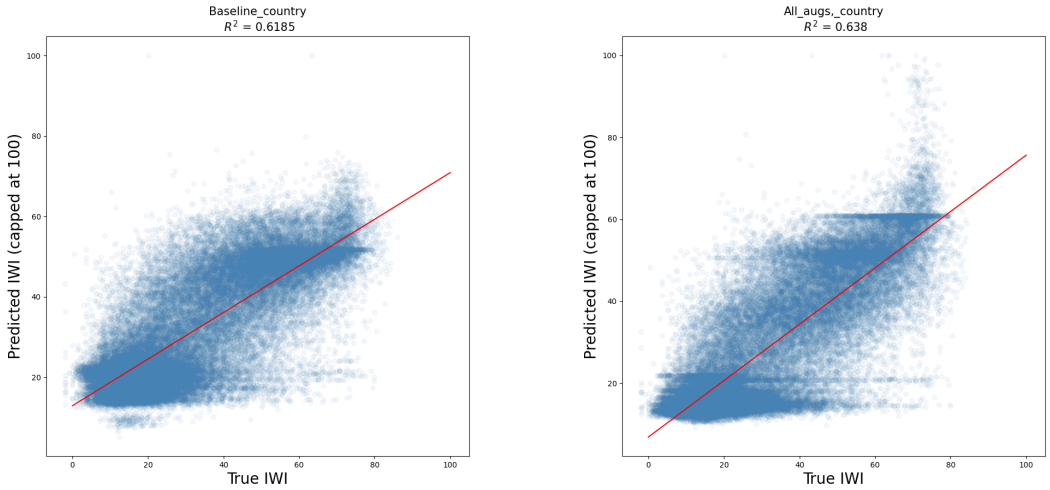
7.3 Performance with country folds evaluation

For the in-country folds, the r^2 values for each of the proposed models can be seen in Table 7.3. Recall that the country folds are used to evaluate the models' abilities to predict on countries it has not trained on, and are primarily geared towards testing the spatial pretrained models. Here, the best performing model is the standard MoCo model with regular augmentations, with neither spatial nor temporal augmentations, and not any of the spatial models which one would expect. The next best performing model is the spatiotemporal model, which also manages to outperform the baseline very slightly.

Model	r^2
All augmentations	0.6380
Temporal	0.5219
Temporal Aug	0.6091
Temporal Aug (no ImageNet)	0.5015
Spatial	0.5859
Spatial Aug	0.5455
Spatiotemporal	0.5962
Spatiotemporal Aug	0.6231

Table 7.3: Results on country folds in r^2.

In Figure 7.2, the best performing proposed model on country folds is compared to the baseline. Furthermore, it depicts a ca. 2 percentage point increase in r^2.



(a) **(b)**

Figure 7.2: Results from baseline model on the **a) in-country** folds and the best performing model **b)** which is a model pre-trained on ImageNet with all augmentations.

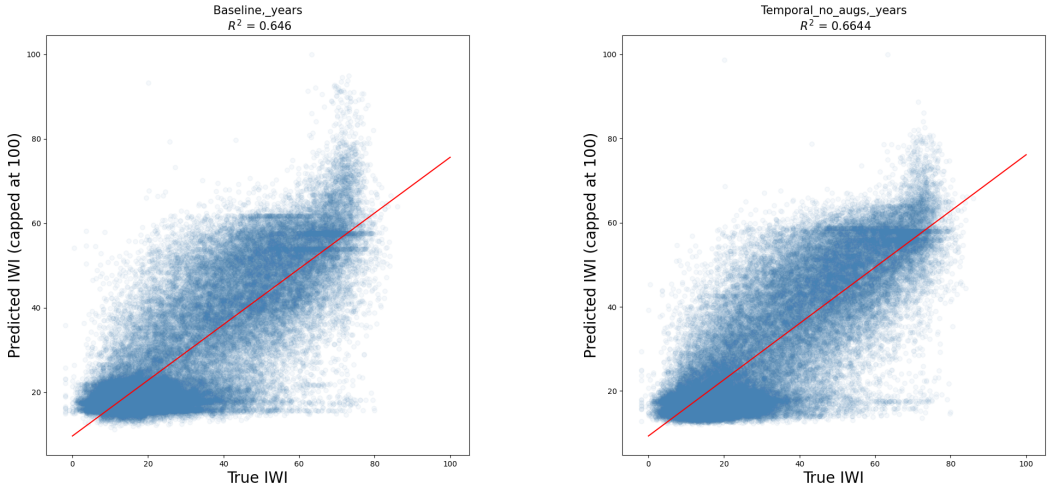
7.4 Performance with out-of-year folds

Again, for the out-of-year folds, the r^2 values for each of the proposed models is presented in Table 7.4. The temporal model with regular augmentations but without being initialized with ImageNet weights before MoCo pretraining was not evaluated on the out-of-year folds since it had shown strictly worse performance compared to its counterpart with ImageNet weights. Here we can note a slight increase, ca. 2 percentage point in performance compared to the baseline model. The model with the best performance was a temporal model without any applied regular augmentations. The temporal model with regular augmentations and the spatial model also outperformed the baseline model.

Model	r^2
All augmentations	0.5942
Temporal	0.6644
Temporal Aug	0.6566
Temporal Aug (no ImageNet)	N/A
Spatial	0.6547
Spatial Aug	0.6172
Spatiotemporal	0.6399
Spatiotemporal Aug	0.6183

Table 7.4: Results on out-of-year folds in r^2.

In Figure 7.3, the best performing model is compared to the baseline. Note that the best performing model outperforms the baseline by a ca. 2 percentage point increase in r^2.



(a) **(b)**

Figure 7.3: Results from baseline model on the **a) out-of-year** folds and the best performing model **b)** which is a model pre-trained on ImageNet with no augmentations.

7.5 Model performance on Urban vs. Rural clusters

Based on the results on the in-country folds, the model performance is also measured in r^2 on the urban and rural clusters separately which can be seen in Figure 7.4. It can be noted that in general, both models are better at predicting poverty for rural clusters compared to urban clusters. The best performing proposed model on the

in-country folds is performing slightly better on rural clusters and noticeably better on urban clusters. Hence, the proposed model is better at distinguishing between wealth in both urban and rural areas, but more prominently in the urban clusters.

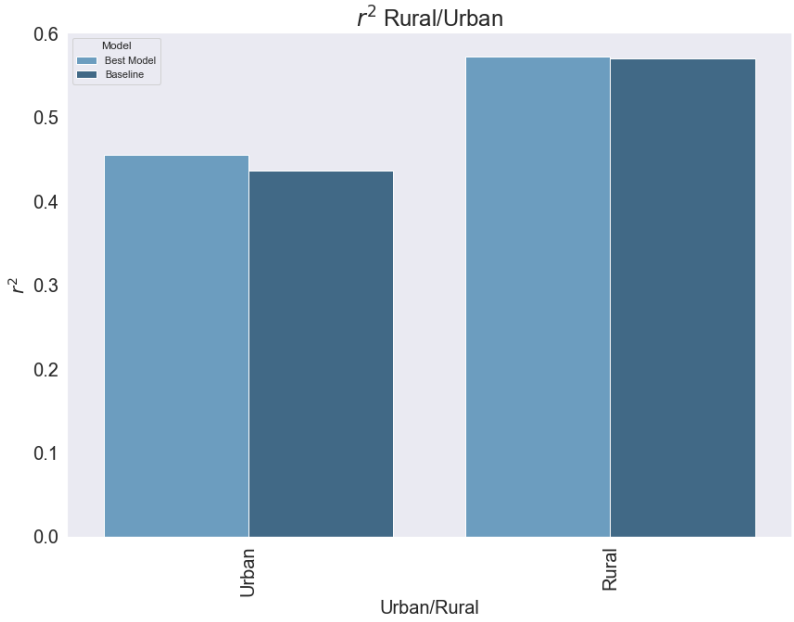
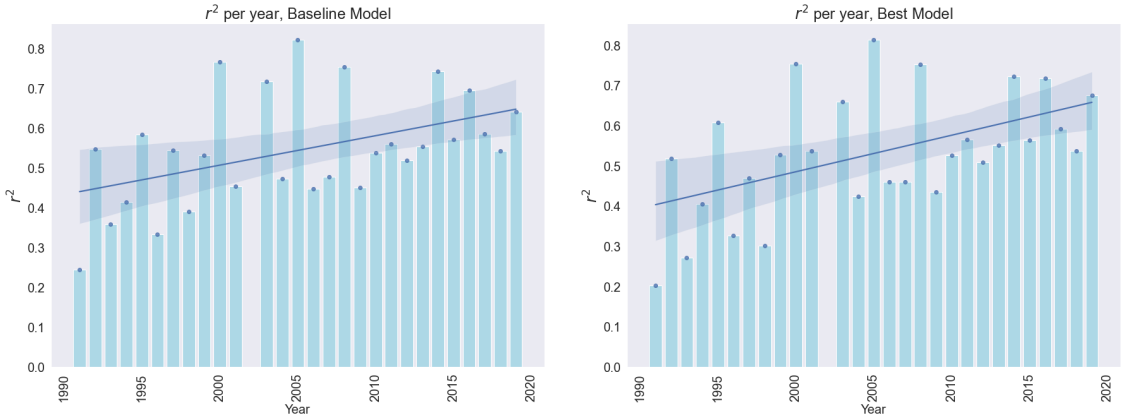


Figure 7.4: The r^2 value of the baseline model and the best performing model on in-country folds. The proposed model performs slightly better on both urban and rural clusters.

7.6 Per-year performance

The models were also evaluated on their proficiency at predicting the poverty index for each year of the surveys, which can be seen in Figure 7.5. It can be observed that the baseline and the best SSL model have very similar r^2 for almost all years and that both increase in performance over years. This is likely explained by the fact that the number of surveys conducted per year also increases over time, as was previously shown in Figure 5.5. Further, the best SSL model actually improves slightly more over time than the baseline counterpart, as the regression line has a steeper slope than that of the baseline.

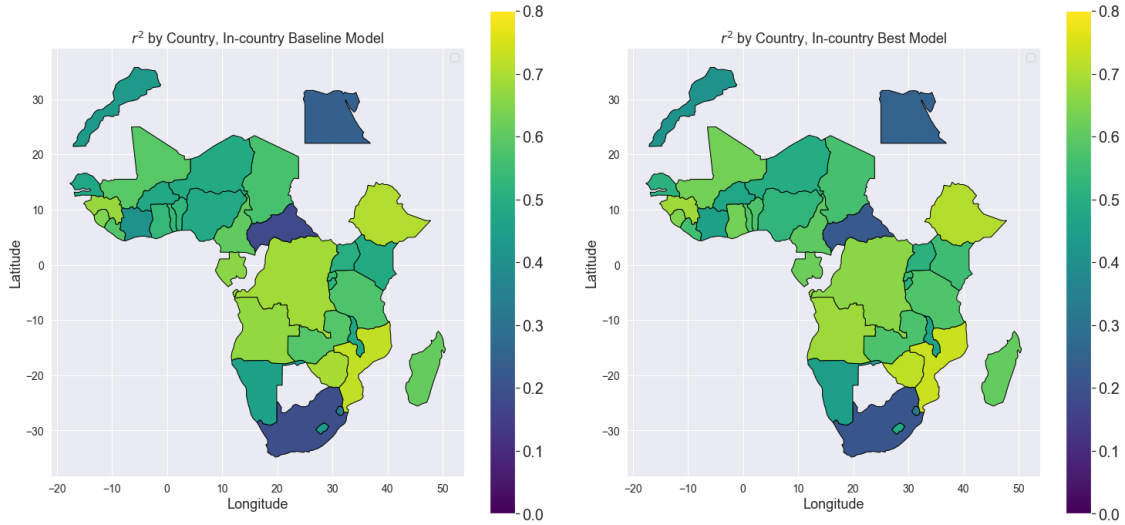


(a) **(b)**

Figure 7.5: Model performance for the different years with a fitted regression line.

7.7 Prediction performance per country

Aside from the general performance across all surveyed countries, an investigation was made to understand how the models predict per country. It can be seen that the proposed model as well as the baseline model performs especially poorly on Egypt, South Africa, and the Central African Republic. Recall the wealth asset index distribution from 5.6, where we can see that both Egypt and South Africa have high median IWI compared to the total median, which could explain the poor results.

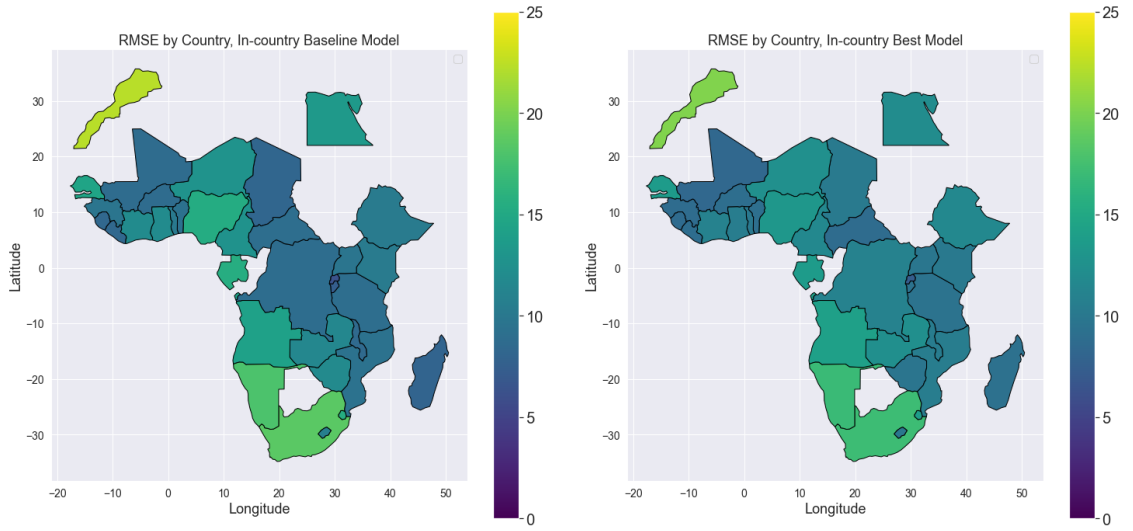


(a) **(b)**

Figure 7.6: **a)** baseline model r^2 per country. **b)**, best performing model on in-country folds r^2 per country.

7.8 Prediction deviation of models

On the other hand, when observing the RMSE values in Figure 7.7, i.e. the average prediction deviations from the true IWI-values, one can note that some countries have skewed RMSE values compared to their r^2. Morocco presents a r^2 of 0.42 but it has the highest RMSE of approximately 22.5, which means that the model, on average, predicts wrong by 22.5 IWI units. A country with the same r^2, like Burkina Faso, presents a much lower RMSE at approximately 9.3 IWI units. However, Morocco also presents a comparatively low number of clusters, which could mean the model haven't learned the relevant information to make an accurate prediction and instead could rely on the average prediction of the whole data set.



(a) **(b)**

Figure 7.7: a), baseline model RMSE per country. **b)**, best performing model on in-country folds RMSE per country.

8

Discussion

This chapter compares the proposed models with the baseline models and discusses the effects of Self-supervised learning on a poverty prediction task. Additionally, preprocessing and processes regarding model implementations are treated. Suggestions for future work are as well presented.

8.1 Comparison of models against baseline

As shown in the results, no single model is the best at all evaluation folds. In fact, different models have the highest r^2 for the three respective sets of folds. For the in-country folds, which divide the data without respect to countries and years in which the survey was conducted, and should give the most general evaluation, the best performing model was the temporal model with regular augmentations, and the spatial model without regular augmentations also outperformed the baseline. It is noteworthy that the temporal model with regular augmentations and the spatial model are the only models to outperform the baseline on two of the evaluation sets.

As for the held out folds, one would expect one of the temporal models to perform best on the held out years folds, and one of the spatial models to perform best on the held out country folds since they are pretrained to be more generalized across the temporal and spatial dimensions respectively. For the held out country folds, the spatial models performed slightly worse than the baseline, despite having spatial pretraining. The best performing model was the default MoCo model with only regular augmentations, which together with the spatiotemporal model with regular augmentations, were the only models outperforming the baseline model. Interestingly, the country folds are also the only folds where either of these models outperforms the baseline.

For the held out years folds, both temporal models outperform the baseline as one would expect, proving that temporal pretraining can in fact boost the performance of an entirely supervised model. Interestingly, the spatial model also outperforms the baseline, however, not by a large margin. Given that DHS surveys are extremely rarely conducted at the same cluster location across years, holding out a set of years also means holding out a set of separate locations. Not only that, but surveys are

also not conducted in each of the 36 countries every year. For example, all surveys collected in Morocco are collected in 2003 and 2004, meaning they are all included in the timespan 2002-2004, and thus in the same fold. The same problem is present for Eswatini, where all surveys are conducted in 2006 and 2007, and the entire country is thus also included in the same fold. This means that entire countries are also held out in the held out year folds, making the evaluation also test the ability to extrapolate on unseen countries. This might explain why the spatial model also performs better than the baseline on the held out year folds, however, this fact clashes with the fact that the spatial model did not outperform the baseline on the country folds.

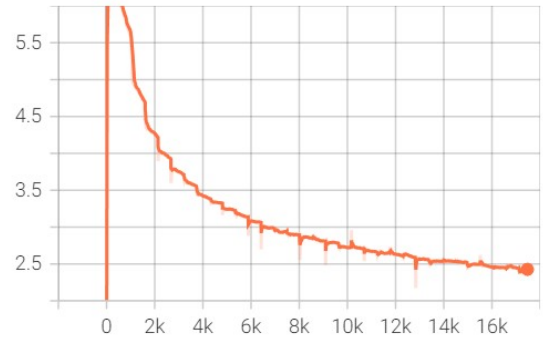
If the SSL pretraining were to do absolutely nothing, one would expect that any model would give roughly the exact same r^2 and that any model would slightly outperform the baseline 50% of the time. Given this fact, the models need to substantially outperform the baseline in order for one to be certain that the reason for the improvement was the pretraining and not simply random chance, which is not the case for all models outperforming the baseline. However, the best performing model on the in-country folds outperforms the baseline model by 2.7 percentage points in r^2, which is a 4.2% increase in explained variance, which should be more than enough to be considered a fluke.

8.2 MoCo pretraining

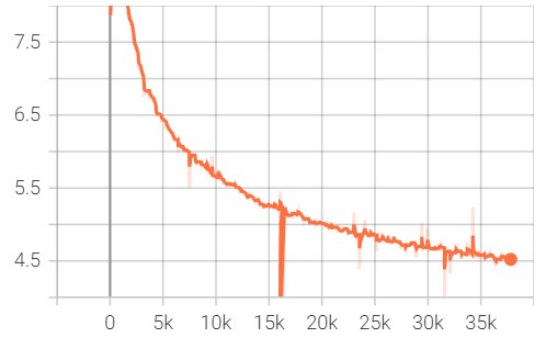
The training time for the MoCo pretext task is very resource- and time-consuming, as one epoch of training takes roughly 4h 40min with distributed learning on four high-end A100 GPUs. This fact, along with constraints on time, means that the amount of SSL pretraining performed for the semi-supervised models differs slightly between models. Some models are pretrained more than others, which could explain some variance in the results of the semi-supervised models. The number of trained epochs per model can be seen in Table 8.1. One can see that the temporal model with regular augmentations has trained the most epochs out of all models, with the reason being that it was the first model to be implemented. In total, it has trained for almost 14 days. The spatiotemporal models took almost twice as long to train compared to the other models and were therefore not able to be trained as long. However, the loss of all models, except for the spatiotemporal models, had more or less stagnated by the time training ended, as can be seen in Figure 8.1, which depicts the training losses of all models with regular augmentations. With more time, the models would have been pretrained more, and for the same number of epochs in order to be able to attribute differences in results entirely to the pretext task.

Model	Epochs trained
Aug	32
Temporal	31
Temporal Aug	70
Temporal Aug (no ImageNet)	63
Spatial	44
Spatial Aug	44
Spatiotemporal	22
Spatiotemporal Aug	22

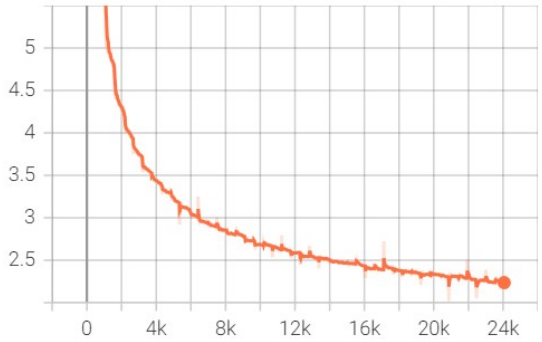
Table 8.1: Number of epochs of MoCo pretraining for each model.



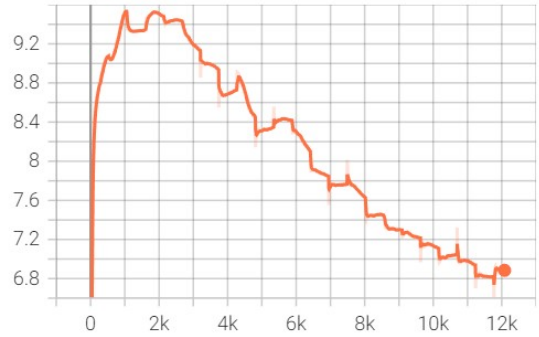
(a) Aug model



(b) Temporal Aug model



(c) Spatial Aug model



(d) Spatiotemporal Aug model

Figure 8.1: Training losses for all models with regular augmentations. Y-axis depicts the contrastive loss, while the x-axis depicts the number of batches.

8.3 Satellite imagery

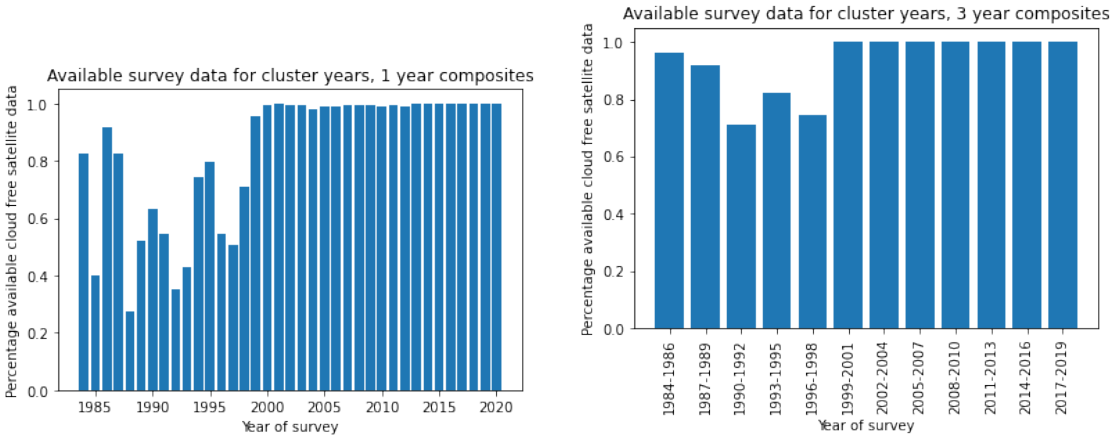
As mentioned in Section 5.1, the satellite project Landsat was selected to be used as input data to both the self-supervised and supervised models. The reasoning for choosing Landsat, despite its relatively poor spatial resolution, was that it has been in orbit for a long time, allowing the use of all available DHS surveys to training the model. Given that the results presented earlier indicate that the models have

increased in performance from self-supervised pretraining, it could be interesting to investigate whether the self-supervised models also decrease the amount of data needed to achieve the same results as entirely supervised models. This was the intention at the start of the project, i.e. to train models on fractions of the dataset and compare them against a baseline. However, this was cut out due to time constraints and the number of combinations of pretraining settings, folds, and fractions becoming unwieldy. If performance on par with a supervised model on 100% of the dataset could be achieved with a fraction of the supervised dataset, one could make the case that the limited time period of the Sentinel satellite project, which has a far superior spatial resolution, would not be a problem. However, the time period of the Sentinel not only puts a bottleneck on the supervised data, but also on the self-supervised data. This means that the same pretraining data, which for Landsat was shown to increase the performance of the baseline model, would not be available to pretrain the model. Arguably, the choice of satellite is largely an empirical question, and would have to be investigated on its own.

Another aspect of the satellite imagery worth discussing is the choice to use three-year composites. This choice was primarily based on a couple of reasons, the main reason being that it has been used in previous work such as Yeh et al. and Pettersson et al. [8, 10], providing the ability to directly compare the model implementations. Using composite imagery should decrease the effects of seasonality since only the median image is used, and increase the fraction of cloud-free images for all locations since there is a lesser likelihood that clouds remain over a specific place for a long period of time. Given that the temporal self-supervised models employed in this project aim to pretrain the model to learn to generalize over the dimension of time, it might be valuable to have median images composed from a smaller time period, say, one year. This would not only give the model up to three times more data to learn from compared to the three-year composites but would also mean images would depict more gradual and well-defined changes over time since they are "medianized" over a smaller timeframe. This could give the model more amounts and more granular information to learn from.

Although a smaller timeframe would lead to more available images per location, it would also mean a higher fraction of locations and timeframes in locations to be unavailable due to clouds in images. This would not put any substantial limitations on the self-supervised data, as more images are available per location, but would decrease the amount of supervised data that could be used. Pettersson et al. [10] calculated the availability of cloud-free images for one-year and three-year composites for survey locations, see Figure 8.2. As can be seen, using three-year composites increases the lowest fraction from 0.3 to roughly 0.7 compared to one-year composites. It is however noteworthy that there is barely any difference in availability between three-year and one-year composites after the year 1999. Years prior to 1999 make up roughly 8.3% of the available surveys used for training, and thus using one-year composites would not decrease the total amount of data for training substantially, but would increase the amount of self-supervised data drastically. With more time, the performance of the models on one-year composites would have been interesting

to investigate.



(a) **(b)**

Figure 8.2: Availability of cloud-free images of survey locations for one-year and three-year composites [10].

8.4 Hyperparameters

When training neural networks, there are numerous hyperparameters to be tuned to maximize the performance of the model, such as e.g. learning rate and choice of optimizer. Given the already large number of combinations of settings and models due to the number of pretext tasks and different evaluation folds, extensive tuning of hyperparameters were not prioritized. Instead, for supervised training, the hyperparameters were the same as those used by Yeh et al. and Pettersson et al. [8, 10], and for the self-supervised training with MoCo, the models were trained with the same hyperparameters as Ayush et al., such as a cosine learning rate schedule [6]. The self-supervised models developed in this project are also primarily compared to the baseline model, and the main focus is to highlight whether the self-supervision methods can improve upon the baseline rather than tuning the best model to obtain a maximum r^2.

8.5 Input image area size

As described in previous sections, DHS applies some noise to the locations where surveys have been conducted in order to protect the confidentiality of respondents. This displacement is up to 2 km for urban clusters, and up to 10 km for the urban clusters. Given that the side length of the images in this project, 6.72 km, is smaller than the maximum potential displacement, it is possible that a rural cluster is located entirely outside of the captured image. However, only 1% of rural clusters are displaced between 0-10 km, and the rest are displaced 0-5 km, meaning this is the case for very few images. This results in a trade-off between including enough

area such that the surveyed cluster is included in the picture, but not enough area to include completely separate areas where the wealth levels might be entirely different. Jean et al. [9] use an input size of 10 km to account for the maximum displacement, whereas Yeh et al. and Pettersson et al. use 6.72 km. Since Yeh et al. and Pettersson et al. have provided better results compared to Jean et al., an input area size of 6.72 x 6.72 km was used.

9

Conclusion

This work presents the possibilities of improving machine learning models aimed at predicting asset wealth on a local level in Africa. Thus the focus of this thesis is to create an outline for which machine learning approaches that can benefit other poverty prediction models. In contrast to prior research, the aim is not to find a final model that surpasses all the predecessed models by earlier works, instead one aims to find a way which can actually improve their works as well through the help of self-supervised learning.

It can be concluded that models that are pre-trained on a relevant pretext task in fact improve predictive accuracy in the downstream task of creating poverty maps. The proposed models in this thesis that have been pre-trained on large unlabelled and freely available remote sensing data sets with a pretext task geared towards poverty prediction did improve the predictive accuracy by approximately $2 - 3\%$ on all evaluation procedures. Although they need different experimental set-ups to accomplish this, it's still a reasonable conclusion to make. This means that for any models that is designated to predict poverty using satellite imagery, it could be worth investigating whether pre-training the model using any of the self-supervised learning approaches presented in this thesis could improve the predictive accuracy more than relying solely on the learned representations from ImageNet as a foundation.

The results also suggest that pretext tasks more inclined on exploiting the temporal aspect of satellite imagery, improve the model predictions comparatively more than a model focused on the spatial or the mix of both.

Bibliography

[1] UNCTAD, "Facts and figures," 2021. [Online]. Available: https://unctad.org/press-material/facts-and-figures-7

[2] M. Yusuf and M. Yusuf, "Community targeting for poverty reduction: lessons from developing countries," 2010. [Online]. Available: https://open.bu.edu/handle/2144/22669

[3] S. O. Rutstein, "The dhs wealth index: Approaches for rural and urban areas, demographic and health research." 2008. [Online]. Available: https://dhsprogram.com/publications/publication-cr6-comparative-reports.cfm

[4] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton, "Big self-supervised models are strong semi-supervised learners," *CoRR*, vol. abs/2006.10029, 2020. [Online]. Available: https://arxiv.org/abs/2006.10029

[5] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," *CoRR*, vol. abs/2002.05709, 2020. [Online]. Available: https://arxiv.org/abs/2002.05709

[6] K. Ayush, B. Uzkent, C. Meng, K. Tanmay, M. Burke, D. B. Lobell, and S. Ermon, "Geography-aware self-supervised learning," *CoRR*, vol. abs/2011.09980, 2020. [Online]. Available: https://arxiv.org/abs/2011.09980

[7] C. Agastya, S. Ghebremusse, I. Anderson, C. Reed, H. Vahabi, and A. Todeschini, "Self-supervised contrastive learning for irrigation detection in satellite imagery," *CoRR*, vol. abs/2108.05484, 2021. [Online]. Available: https://arxiv.org/abs/2108.05484

[8] C. Yeh, A. Perez, Driscoll, and A. et al., "Using publicly available satellite imagery and deep learning to understand economic well-being in africa." *Nat Commun*, vol. 11, no. 2583, 2020. [Online]. Available: https://www.nature.com/articles/s41467-020-16185-w

[9] N. Jean, M. Burke, M. Xie, W. M. Davis, D. B. Lobell, and S. Ermon, "Combining satellite imagery and machine learning to predict poverty,"

Science, vol. 353, no. 6301, pp. 790–794, 2016. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.aaf7894

[10] J. Ortheden and M. Pettersson, "Predicting economic well-being in africa using temporal satellite imagery and deep learning," 2021. [Online]. Available: https://drive.google.com/file/d/1rv2xezwEr3w7OfQw7AZqiN-KtWQmUwqu/view

[11] M. Pinkovskiy and X. Sala-i Martin, " Lights, Camera . . . Income! Illuminating the National Accounts-Household Surveys Debate *," *The Quarterly Journal of Economics*, vol. 131, no. 2, pp. 579–631, 02 2016. [Online]. Available: https://doi.org/10.1093/qje/qjw003

[12] X. Chen and W. D. Nordhaus, "Using luminosity data as a proxy for economic statistics," *Proceedings of the National Academy of Sciences*, vol. 108, no. 21, pp. 8589–8594, 2011. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.1017031108

[13] J. V. Henderson, A. Storeygard, and D. N. Weil, "Measuring economic growth from outer space," *American Economic Review*, vol. 102, no. 2, pp. 994–1028, April 2012. [Online]. Available: https://www.aeaweb.org/articles?id=10.1257/aer.102.2.994

[14] N. Gorelick, M. Hancher, M. Dixon, S. Ilyushchenko, D. Thau, and R. Moore, "Google earth engine: Planetary-scale geospatial analysis for everyone," *Remote Sensing of Environment*, vol. 202, pp. 18–27, 2017, Big Remotely Sensed Data: tools, applications and experiences. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425717302900

[15] F.-C. Hsu, K. E. Baugh, T. Ghosh, M. Zhizhin, and C. D. Elvidge, "Dmsp-ols radiance calibrated nighttime lights time series with intercalibration," *Remote Sensing*, vol. 7, no. 2, pp. 1855–1876, 2015. [Online]. Available: https://www.mdpi.com/2072-4292/7/2/1855

[16] C. D. Elvidge, K. Baugh, M. Zhizhin, F. C. Hsu, and T. Ghosh, "Viirs night-time lights," *International Journal of Remote Sensing*, vol. 38, no. 21, pp. 5860–5879, 2017. [Online]. Available: https://doi.org/10.1080/01431161.2017.1342050

[17] C. Kone, "Introducing convolutional neural networks in deep learning," *Towards Data Science*, 2022. [Online]. Available: https://towardsdatascience.com/introducing-convolutional-neural-networks-in-deep-learning-400f9c3ad5e9

[18] I. S. Mohamed, "Convolutional neural network," *ResearchGate*, 2022. [Online]. Available: https://www.researchgate.net/figure/An-example-of-convolution-operation-in-2D-2_fig3_324165524

[19] A. Zhang, "Residual networks (resnet)," *Dive into Deep Learning*, 2022. [Online]. Available: https://d2l.ai/chapter_convolutional-modern/resnet.html

[20] N. Harris, "Visualizing dbscan clustering," 2015. [Online]. Available: https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

[21] X. Chen, H. Fan, R. Girshick, and K. He, "Improved Baselines with Momentum Contrastive Learning," *arXiv e-prints*, p. arXiv:2003.04297, Mar. 2020.

[22] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *CoRR*, vol. abs/1807.03748, 2018. [Online]. Available: http://arxiv.org/abs/1807.03748

[23] N. E. Young, R. S. Anderson, S. M. Chignell, A. G. Vorster, R. Lawrence, and P. H. Evangelista, "A survival guide to landsat preprocessing," *Ecology*, vol. 98, no. 4, pp. 920–932, 2017. [Online]. Available: https://esajournals.onlinelibrary.wiley.com/doi/abs/10.1002/ecy.1730

[24] J. Smits and R. Steendijk, "The international wealth index (iwi)," *Social Indicators Research*, vol. 122, pp. 65–85, 2014. [Online]. Available: https://doi.org/10.1007/s11205-014-0683-x

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

A

Appendix

A.1 Countries in survey data set

Country	Number of clusters	Mean IWI	Country	Number of clusters	Mean IWI
Angola	969	32.68	Madagascar	1743	19.57
Benin	1710	28.39	Malawi	3184	19.68
Burkina Faso	1740	24.15	Mali	1941	26.47
Burundi	1128	19.03	Morocco	476	54.65
Central African Republic	225	16.23	Mozambique	1136	27.34
Cameroon	1613	35.35	Namibia	1301	37.43
Chad	624	16.46	Niger	330	18.22
Comoros	242	38.37	Nigeria	4074	36.33
Cote D'Ivoire	725	33.11	Rwanda	1685	20.64
Democratic Rep. of Congo	783	18.43	Senegal	2165	35.37
Egypt	7638	59.50	Sierra Leone	1678	24.23
Ethiopia	2227	17.25	South Africa	746	58.23
Gabon	332	41.42	Swaziland	270	38.00
Ghana	2325	34.98	Tanzania	2668	23.24
Guinea	1273	29.12	Togo	773	28.69
Kenya	2616	26.67	Uganda	1947	25.36
Lesotho	1162	26.77	Zambia	1573	27.14
Liberia	772	20.46	Zimbabwe	1401	32.83

Table A.1: Surveyed countries in data set and its corresponding number of clusters as well as mean IWI.

A.2 r^2 plots of in-country folds models.

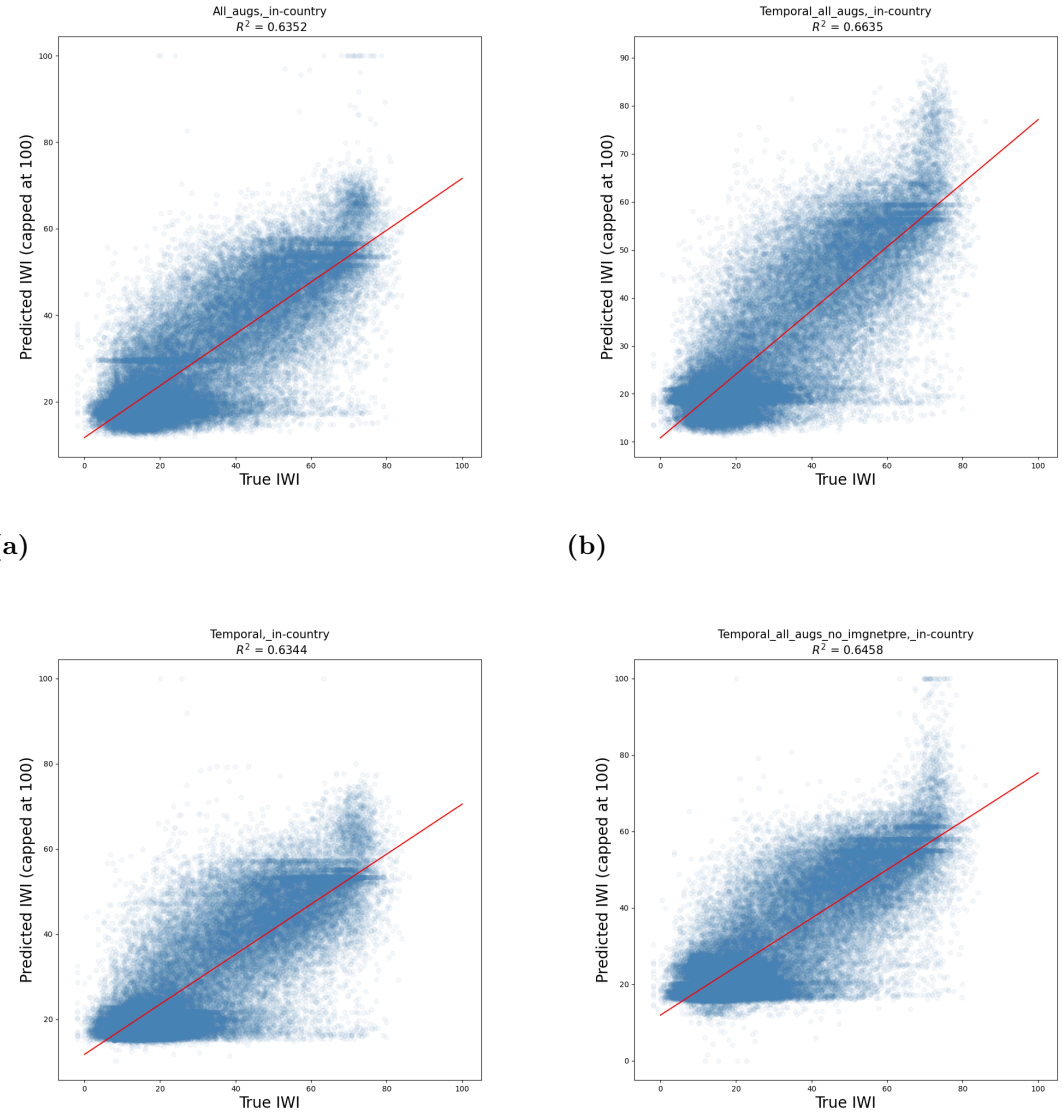
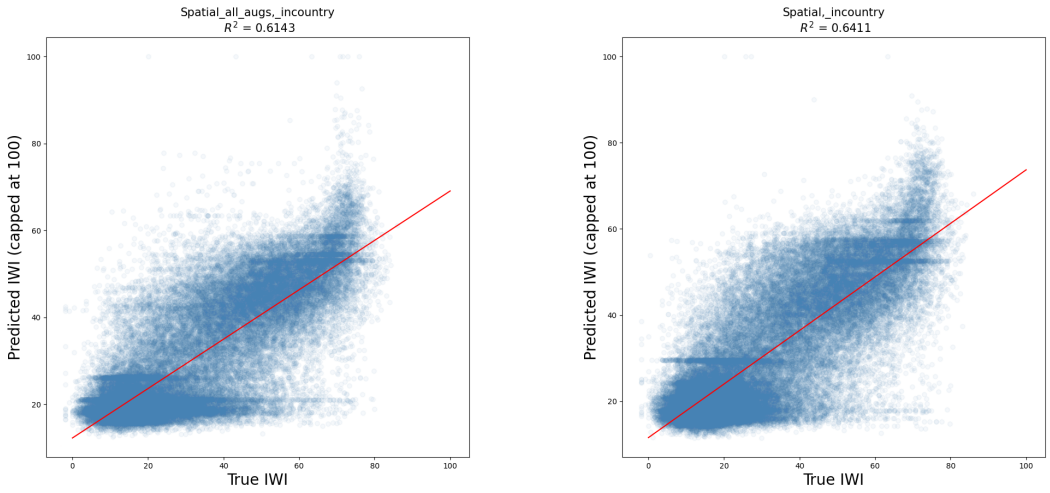
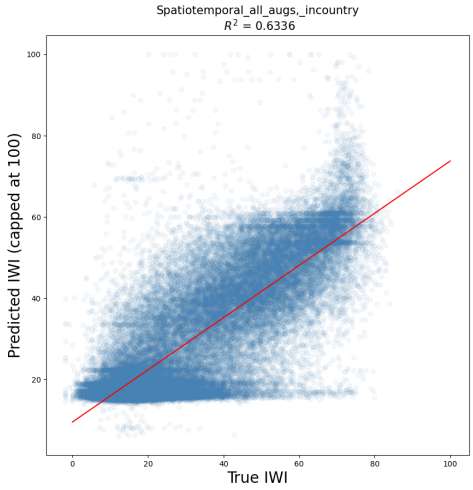


Figure A.1: Results from the temporal and regular augmentation self-supervised models on the **in-country** folds.

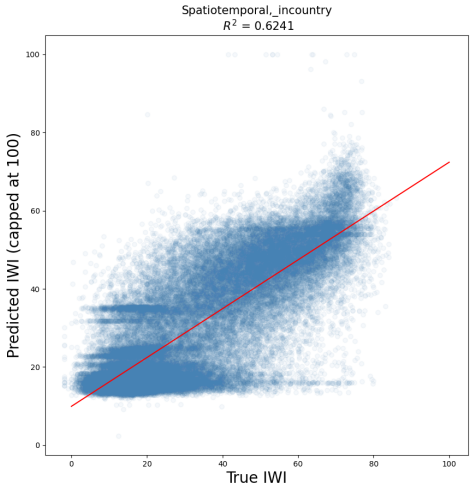


(a)

(b)



(c)



(d)

Figure A.2: Results from the spatial and spatiotemporal self-supervised models on the **in-country** folds.

A.3 r^2 plots of country folds models.

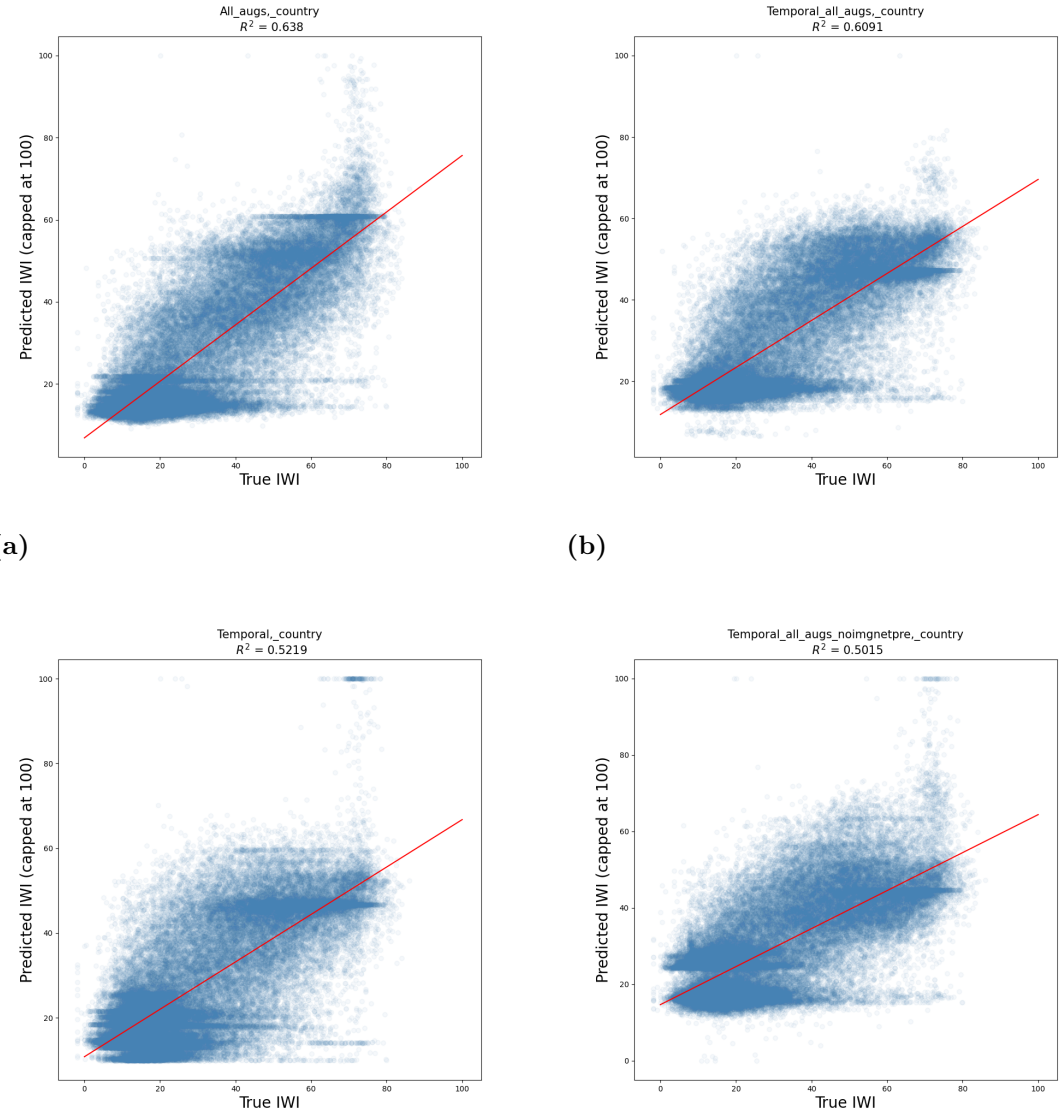
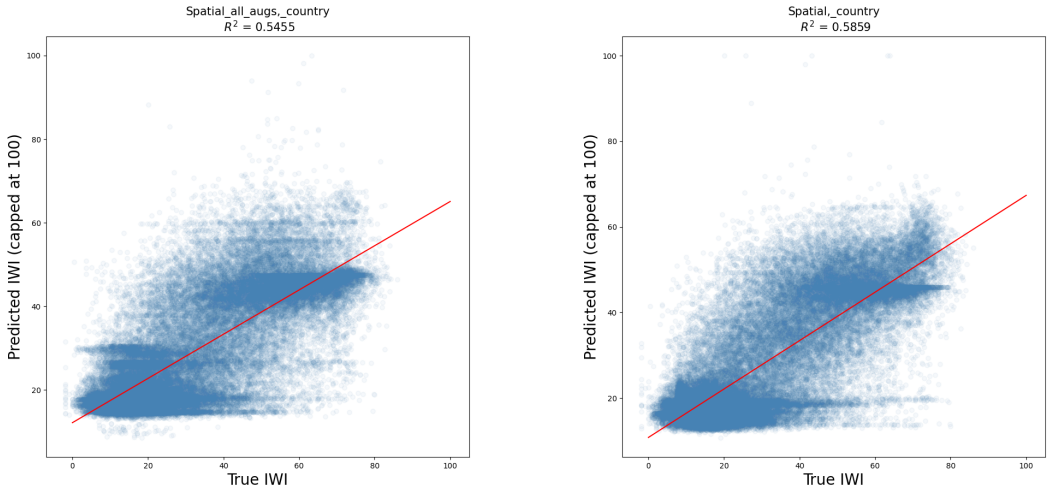


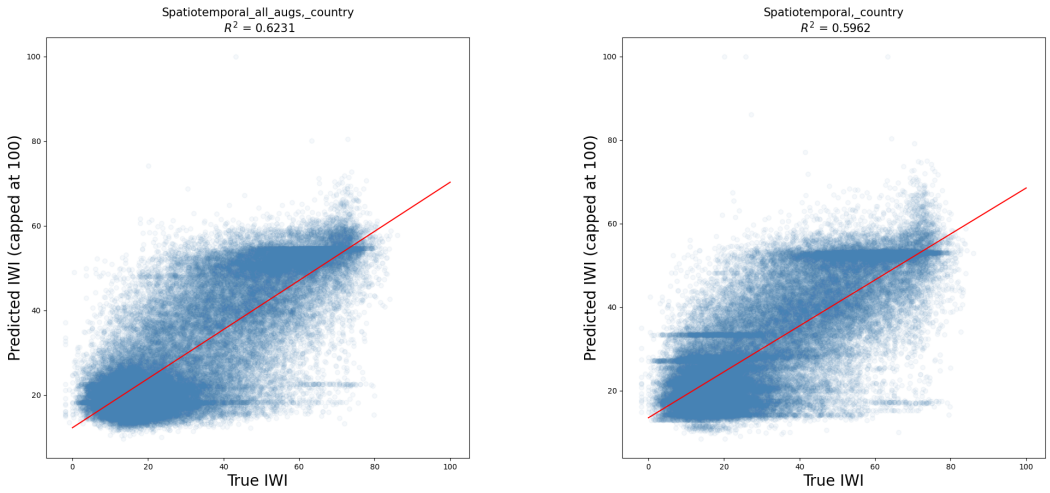
Figure A.3: Results from the temporal and regular augmentation self-supervised models on the **country** folds.

A.4 r^2 plots of country folds models.



(a)

(b)

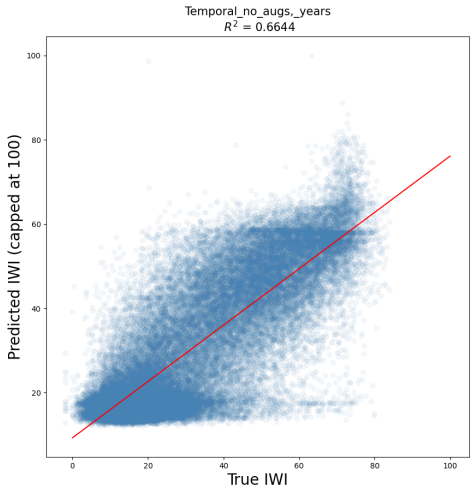


(c)

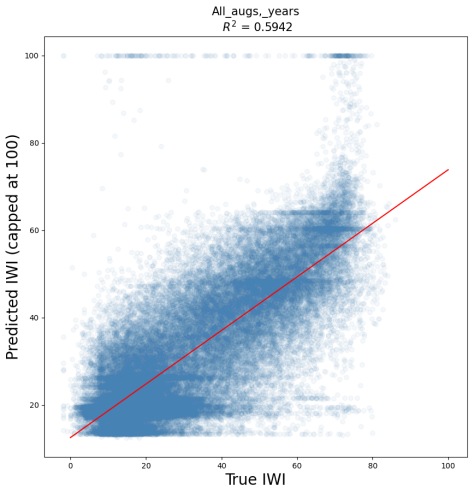
(d)

Figure A.4: Results from the spatial and spatiotemporal self-supervised models on the **country** folds.

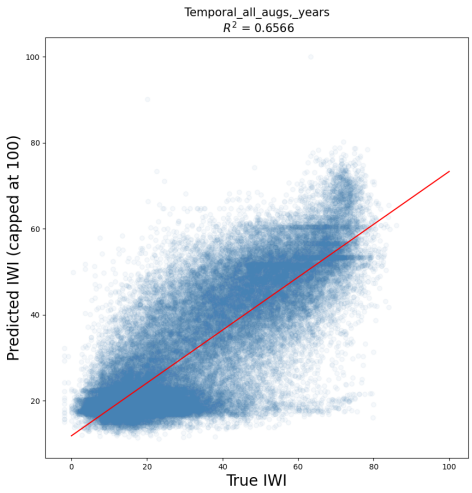
A.5 r^2 plots of out-of-year folds models.



(a)

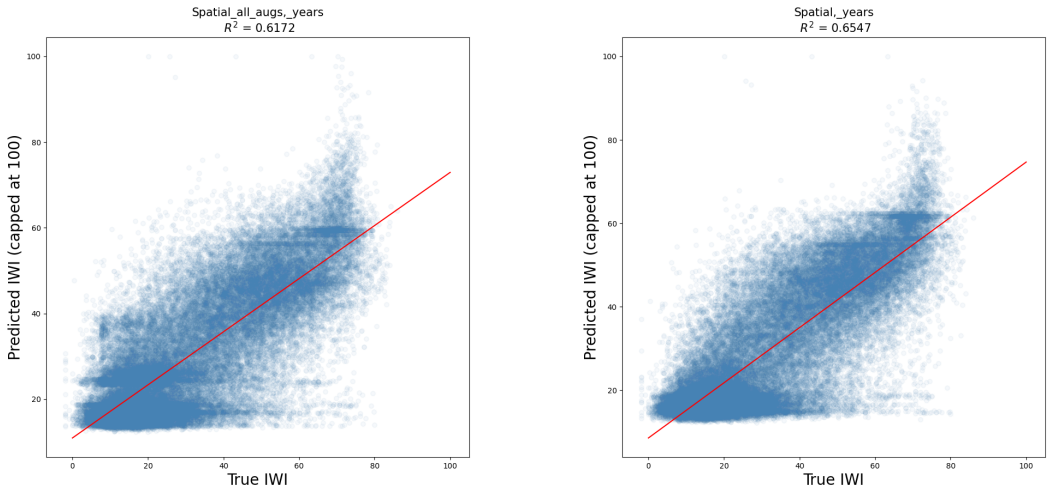


(b)



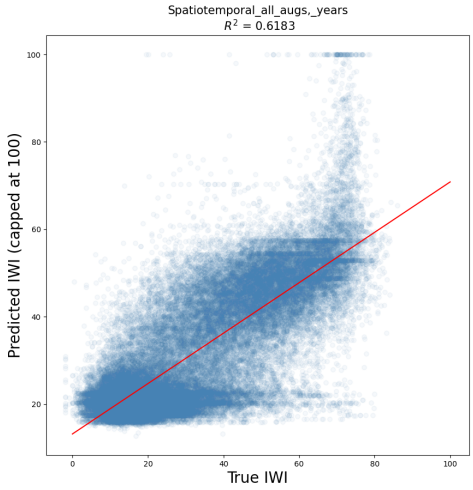
(c)

Figure A.5: Results from self-supervised models on the **out of year** folds.

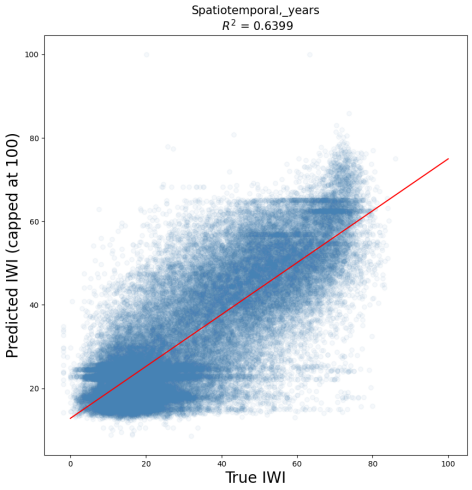


(a)

(b)



(c)

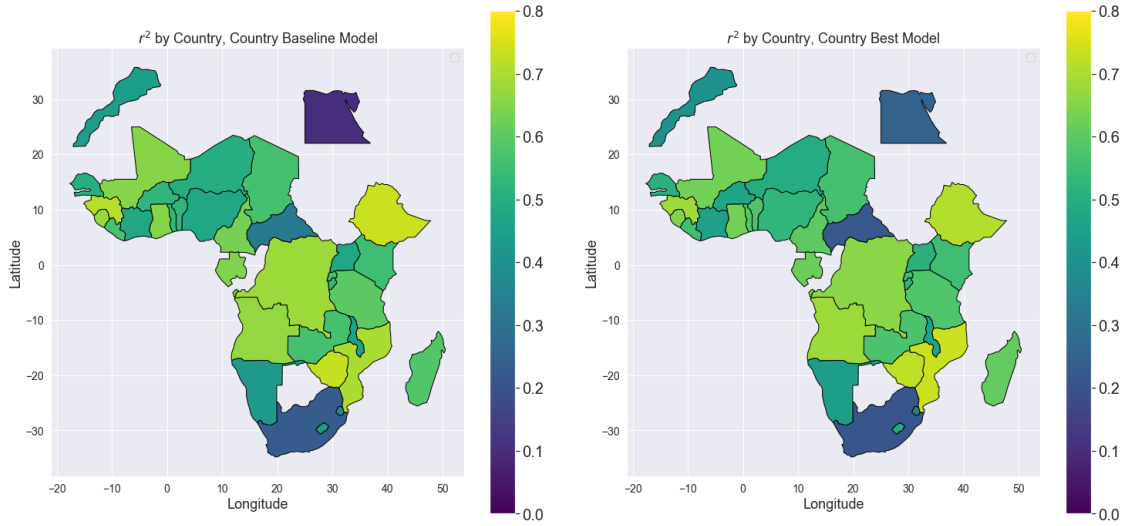


(d)

Figure A.6: Results from the temporal and regular augmentation self-supervised models on the **out of year** folds.

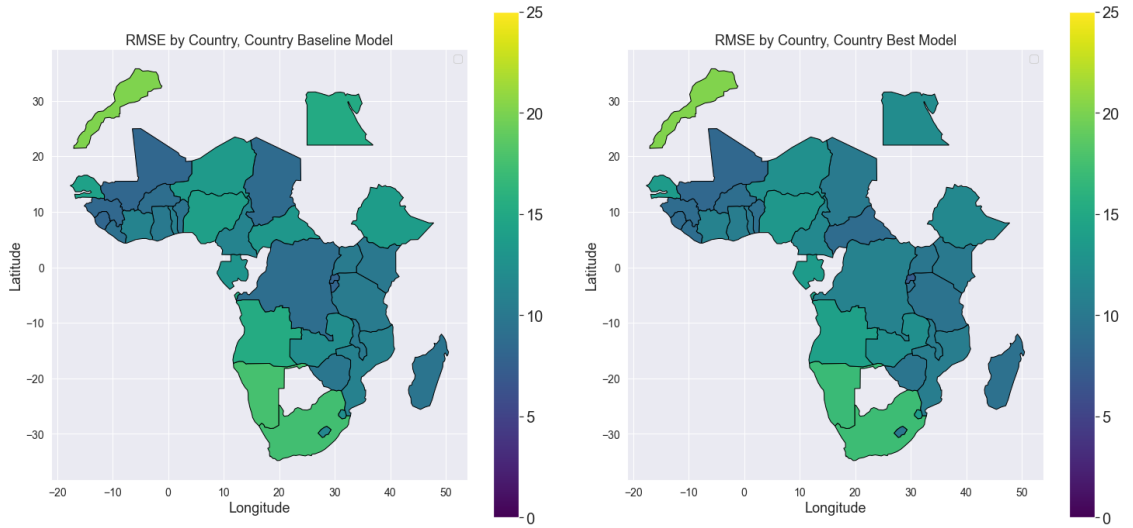
A.6 r^2 and RMSE per country for Country folds and Out-of-year folds.

A.6.1 Country folds



(a) (b)

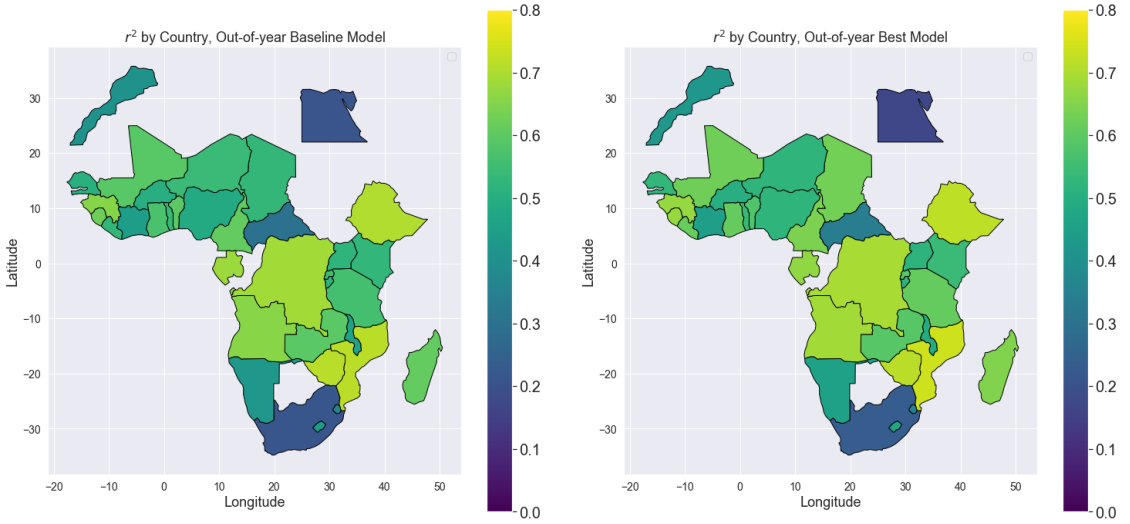
Figure A.7: Results from baseline model on the **a) country** folds and the best performing model **b)** which is a model pre-trained on ImageNet with all augmentations.



(a) (b)

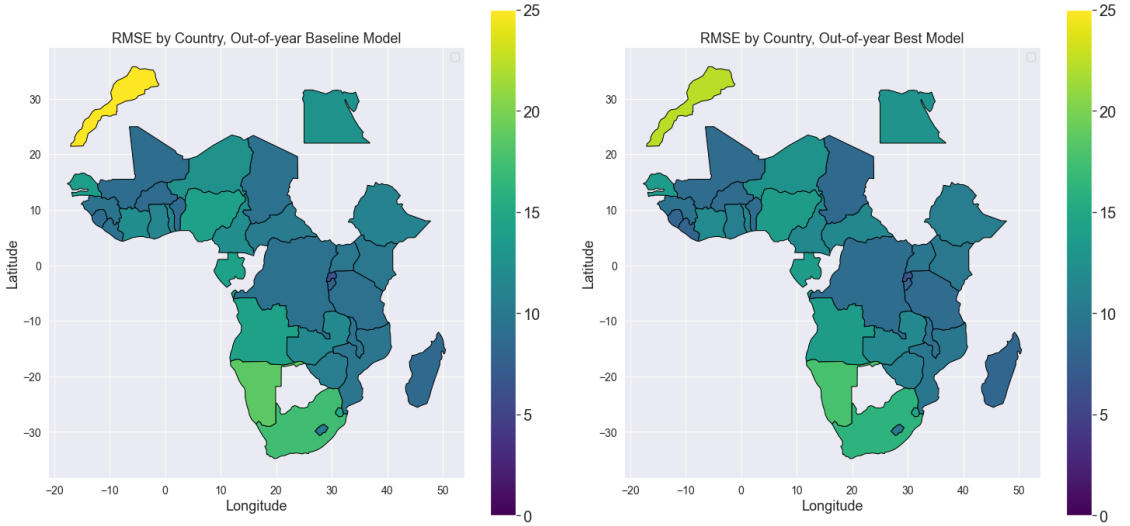
Figure A.8: Results from baseline model on the **a) country** folds and the best performing model **b)** which is a model pre-trained on ImageNet with all augmentations.

A.6.2 Out-of-year folds



(a) **(b)**

Figure A.9: Results from baseline model on the **a) country** folds and the best performing model **b)** which is a model pre-trained on ImageNet with all augmentations.



(a) **(b)**

Figure A.10: Results from baseline model on the **a) country** folds and the best performing model **b)** which is a model pre-trained on ImageNet with all augmentations.