

Technische Universität Berlin

Faculty of Electrical Engineering and Computer Science
Dept. of Computer Engineering and Microelectronics
Remote Sensing Image Analysis Group



Large-scale contrastive self-supervised representation learning for content-based remote sensing image retrieval

Master of Science in Computer Engineering

September, 2021

Clasen, Kai Norman


Matriculation Number: 364047

Supervisor: Prof. Dr. Begüm Demir
Advisor: Gencer Sümbül

Declaration

I hereby confirm to have written the following thesis on my own, not having used any other sources or resources than those listed.

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt habe. Sämtliche benutzten Informationsquellen sowie das Gedankengut Dritter wurden im Text als solche kenntlich gemacht und im Literaturverzeichnis angeführt. Die Arbeit wurde bisher nicht veröffentlicht und keiner Prüfungsbehörde vorgelegt.

Berlin, September 7, 2021 
Kai Norman Clasen

Abstract

This thesis presents a novel metadata-guided sampling framework to improve the content-based image retrieval (CBIR) performance of contrastive self-supervised representation learning (CSSRL) methods. The proposed framework utilizes remote sensing imagery’s freely available location metadata to cluster the data into groups with higher similarity. These clusters are then used to guide the batch assembly process with variable hardness. Assembling a batch from a single cluster increases the batch hardness while sampling all images from different clusters decreases the hardness. The conducted experiments show the effectiveness of utilizing location information to guide the sampling process. Concretely, three CSSRL methods — SimCLR, Barlow Twins, and BYOL — are investigated in detail. The results demonstrate that the CBIR performance of these methods benefits from easier batches. Besides investigating the effect of the proposed sampling framework, the thesis critically evaluates the default augmentation pipeline of these methods and proposes a pipeline tailored for the RS domain. The recommended pipeline consists of resized cropping, rotating, Dihedral transformation, and Gaussian blurring. The validity of the presented pipeline is experimentally verified.

Zusammenfassung

In dieser Arbeit wird ein neuartiges, metadatengeleitetes Samplingverfahren präsentiert, das zur Verbesserung der inhaltsbasierten Bildabfrage (CBIR) von kontrastiven, selbstüberwachten Representation Learning (CSSRL) Methoden verwendet werden kann. Das vorgeschlagene Framework nutzt die Metadaten von frei verfügbaren Satellitenbildern, um die Daten in Gruppen mit höherer Ähnlichkeit zu unterteilen. Diese Cluster werden dann verwendet, um die Batch-Zusammenführung mit variabler Härte umzusetzen. Die Zusammenstellung eines Batches aus einem einzigen Cluster erhöht die Batch-Härte, während die Entnahme aller Bilder aus verschiedenen Clustern die Härte reduziert. Die durchgeführten Experimente zeigen die Wirksamkeit der Verwendung von Standortinformationen, um das Samplingverfahren zu steuern. Konkret werden drei CSSRL-Methoden – SimCLR, Barlow Twins, und BYOL – im Detail untersucht. Die Ergebnisse belegen, dass die CBIR-Leistung dieser Methoden von einfacheren Batches profitiert. Neben der Untersuchung der Auswirkungen des vorgeschlagenen Sampling-Frameworks, wird in dieser Arbeit auch die standardmäßige Augmentierungspipeline dieser Methoden kritisch bewertet und eine auf den RS-Bereich zugeschnittene Pipeline vorgeschlagen. Die empfohlene Pipeline besteht aus Größenanpassung, Rotation, Dihedral-Transformation und Gaußscher Unschärfe. Die Gültigkeit der vorgestellten Pipeline wird experimentell verifiziert.

Contents

List of Figures	xi
List of Tables	xv
1 Introduction to Remote Sensing (RS) Content-Based Image Retrieval (CBIR)	1
2 Self-Supervised Representation Learning (SSRL)	7
2.1 Introduction to SSRL	7
2.2 Contrastive SSRL (CSSRL)	9
3 Proposed Augmentation Pipeline for CSSRL in RS	17
4 Proposed Metadata-Guided Sampling Framework for CSSRL in RS	23
4.1 Motivation	23
4.2 Relation to Similar Work in RS Domain	26
4.3 Location-Guided Clustering	27
4.4 In-Cluster Sampling	29
4.5 Mixed-Cluster Sampling	30
4.6 Key Points of Proposed Metadata-Guided Sampling Framework . .	31
5 Dataset and Experimental Setup	33
5.1 The BigEarthNet-S2 Archive	33
5.2 Evaluation Metrics	36
5.3 Default Experimental Setup	42
6 RS CBIR Results	47
6.1 Analysis of Investigated CSSRL Methods	47
6.2 Analysis of Proposed Metadata-Guided Sampling Framework for CSSRL	60
7 Conclusion	65
A Additional BigEarthNet Resources	75
B Extended Experimental Results	79

List of Figures

1.1	Remote sensing content-based image retrieval example	2
1.2	Supervised Learning Diagram	2
1.3	Example image from the UC Merced dataset shown with single and multi-label annotations.	3
1.4	Unsupervised Learning Diagram	4
2.1	Self-Supervised Representation Learning Diagram	7
2.2	Visual contrastive loss update example on a sphere.	10
2.3	Visualization of positive view generation through image augmentation.	11
2.4	Steps through a general contrastive self-supervised framework.	11
2.5	SimCLR Framework	12
2.6	Barlow Twins Framework	14
2.7	BYOL Framework	15
3.1	Examples of Dihedral transformations	18
3.2	Examples of continuous rotations	19
3.3	Example of Gaussian blurring	19
3.4	Examples of resized cropping	20
3.5	Examples of brightness shifting	20
3.6	Examples of contrast shifting	21
3.7	Examples of saturation shifting	21
3.8	Examples of hue shifting.	22
4.1	Positive pairs example	23
4.2	Location-Guided Sampling Framework	25
4.3	View on earth from space vs. projected map.	28
4.4	In-cluster sampling visualization	30
4.5	Mixed-cluster sampling visualization	31
5.1	Spectral Bands from Sentinel-2 Satellite	34
5.2	Example Patch Visualization	34
5.3	Distribution of BigEarthNet patches.	35
5.4	Visualization of BigEarthNet patches containing specific labels.	36
5.5	CBIR examples with precision = 1	38
5.6	Skip Connection Visualization	42
5.7	BigEarthNet-Summer cluster visualizations with many clusters	44
5.8	BigEarthNet-Summer cluster visualizations with few clusters	45

6.1	Example retrieval results	48
6.2	Default benchmark results with the presented CSSRL methods (Barlow Twins, BYOL, SimCLR), a randomly initialized model, as well as a model trained with supervision.	49
6.3	Simplified multi-spectral color-jittering augmentation (brightness + contrast shifting) results with SimCLR method and different <code>max_lighting</code> values.	50
6.4	Simplified multi-spectral color-jittering augmentation (brightness + contrast shifting) results with Barlow Twins method and different <code>max_lighting</code> values.	51
6.5	Simplified multi-spectral color-jittering augmentation (brightness + contrast shifting) results with BYOL method and different <code>max_lighting</code> values.	51
6.6	Dihedral transformation results with Barlow Twins method.	52
6.7	Effect of rotation and Dihedral transformation on the Barlow Twins method.	53
6.8	Gaussian Blurring results with Barlow Twins method.	53
6.9	Resized cropping results with Barlow Twins method and different minimum crop sizes.	54
6.10	Resized cropping results with BYOL method and different minimum crop sizes.	54
6.11	Intermediate NDCG scores for SimCLR, BYOL, and Barlow Twins during training.	55
6.12	Intermediate NDCG scores for SimCLR with different batch sizes.	56
6.13	Intermediate NDCG scores for SimCLR with default Adam and Ranger21 optimizer.	57
6.14	Simplified multi-spectral color-jittering augmentation (brightness + contrast shifting) results with SimCLR method + Ranger21 optimizer and different <code>max_lighting</code> values.	58
6.15	Resized cropping results with SimCLR method + Ranger21 optimizer and different <code>max_lighting</code> values.	58
6.16	Intermediate NDCG scores for BYOL with default Adam and Ranger21 optimizer.	59
6.17	SimCLR in-cluster sampling results with c clusters and the batch size in parenthesis.	60
6.18	SimCLR mixed-cluster sampling results with c clusters.	61
6.19	Barlow Twins in-cluster sampling results with c clusters and the batch size in parenthesis.	61
6.20	Barlow Twins mixed-cluster sampling results with c clusters.	62
6.21	BYOL in-cluster sampling results with c clusters and the batch size in parenthesis.	62
6.22	BYOL mixed-cluster sampling results with c clusters.	63

6.23	BYOL default sampling with different batch sizes vs. mixed-cluster sampling results with c clusters.	63
A.1	Example images containing the respective classes from the 19 class-nomenclature.	77
B.1	Simplified multi-spectral color-jittering augmentation (brightness + contrast shifting) results with SimCLR method.	79
B.2	Simplified multi-spectral color-jittering augmentation (brightness + contrast shifting) results with Barlow Twins method.	80
B.3	Simplified multi-spectral color-jittering augmentation (brightness + contrast shifting) results with BYOL method.	81
B.4	Dihedral transformation results with BYOL method.	82
B.5	Dihedral transformation results with SimCLR method.	83
B.6	Gaussian Blurring results with SimCLR method.	84
B.7	Gaussian Blurring results with BYOL method.	85
B.8	Resized cropping results with SimCLR method.	86
B.9	Effect of Adam (default) vs. Ranger21 optimizer on Barlow Twins.	87
B.10	Dihedral transformation results with Barlow Twins method + Ranger21 optimizer.	88
B.11	Gaussian Blurring results with Barlow Twins method + Ranger21 optimizer.	89
B.12	Effect of Adam (default) vs. Ranger21 optimizer on BYOL.	90
B.13	SimCLR + Ranger21 + in-cluster sampling with 10 clusters. Displaying NDCG scores over time.	90

List of Tables

4.1	Example Cluster Statistics	29
A.1	Relation between New and Original Class-Nomenclature.	75

1 Introduction to Remote Sensing (RS) Content-Based Image Retrieval (CBIR)

Remote sensing (RS) images are a rich source of information for Earth surface monitoring, permitting the development of various programs and studies. These images can help in urban planning, weather prediction, disaster rescue, and climate change analysis [1–3]. Climate change applications such as glaciers tracking [4], water quality studies [5], and global surface water occurrence mapping [6] use satellites as an RS image source. The utilization of satellite imagery is becoming increasingly popular.

One of the main reasons for the surging popularity of RS applications is the availability of global, cheap, and convenient satellite imagery [7]. Some earth observation (EO) satellite programs provide their data for free, such as the Landsat [8] and the Sentinel series [9]. Tools like the Google Earth Engine [10], Copernicus Open Access Hub [11], and the EarthServer [12] make it easy to interact and download the relevant imagery for further processing. However, it may be hard to find the relevant data, as satellite missions create an abundant amount of data. The Sentinel-2 mission alone can produce over 1.6 TBytes of raw compressed data per day [9]. Therefore, it is crucial to have an easy interface to search for relevant images in these immense archives.

If the relevant data for an EO application is defined by metadata, a simple retrieval system is sufficient. The metadata could be a *by-product*, like the exact location and acquisition date, of the sensed tile, or manually annotated keywords. The system could compress and index the metadata and make it trivial to search for specific locations or keywords.

A more complex retrieval approach is content-based image retrieval (CBIR), where given a query image, similar images are returned, independent of the images' locations. CBIR methods require complex models but are not limited to the simple, or labor-intensive metadata, resulting in the major adoption in the RS IR domain [7].

Instead of relying on metadata-based image retrieval, content-based image retrieval utilizes representation learning-based methods. Representation learning refers to the process of learning a mapping from the input to a lower-dimensional feature vector, which abstracts the input to the essential properties [13]. The feature vector may embed information about the input image's texture, shape, or spectral properties. After converting the query to a feature vector, the resulting vector is compared against all the archive images. The most *similar* images are returned as a retrieval result, based on a method-specific similarity measure.



Figure 1.1: Remote sensing content-based image retrieval example. Image with red border is the query image and the blue bordered ones are the retrieved samples with similar content.

Fig. 1.1 shows a possible retrieval result of a RS CBIR system.

The first prominent RS CBIR methods utilized low-level features to encode the images. Domain experts handcrafted feature descriptors that used texture [14], shape [15, 16], spectral cues [17] or a combination of all these to generate the searchable embeddings [18, 19]. Although hand-crafted for the RS domain, these methods do not perform exceptionally well, as they are sensitive to noise or illumination changes [2].

Inspired by the great success of convolutional neural networks in the computer vision field [20–22], researchers adapted these architectures to the RS CBIR domain, outperforming previous handcrafted low-level feature methods [1–3]. Convolutional neural networks learn to encode high-level discriminative visual features directly from the training data [23]. Although there is no need to handcraft low-level feature descriptors, large amounts of data is necessary to train the model. Most proposed state-of-the-art models require the training data to be manually labeled, as they learn under a *supervised* regime.

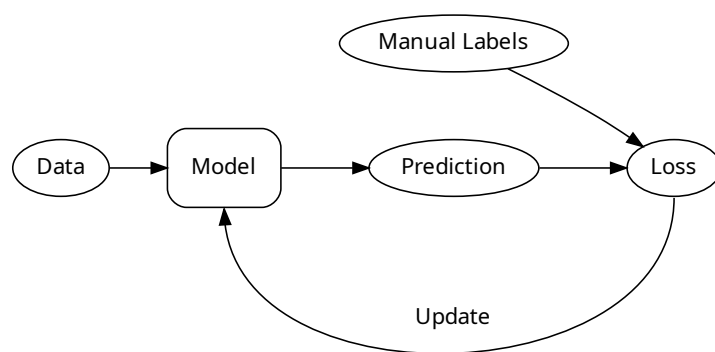


Figure 1.2: Supervised Learning Diagram

The general supervised learning procedure is visualized in Fig. 1.2. First, data is fed to the model, then the model *predicts* an output, for example, the images' category. The prediction of the model is compared to the *correct* labels or *ground-truth*. The loss is then used to tell the model how *wrong* the prediction was and is used to update the model's parameters. This update step is the *learning* part. Supervised learning is domain-independent and works with natural and RS imagery. The main difference regarding the training loop is that natural imagery often uses a single label, whereas RS uses multi-label annotations.

The need for multi-label categorization is due to the high spatial resolution of the images. Each satellite image covers a vast area compared to natural imagery. Although the first labeled RS archives started with a single high-level label [24], most recent RS archives use a multi-label classification scheme to describe the relevant categories for each image [25, 26]. With multi-label data, the model is able to learn fine-grained class-discriminative features. A downside of these fine-grained class-discriminative features is the possible introduction of a bias into the model, as the learned encoders are optimized to be class discriminative and not to capture class-shared or intra-class features well [27]. Fig. 1.3 compares the original high-level single-label [24] against the finer multi-label [25] annotation style of the UC Merced dataset.



(a) Single Label from [24]:
Tenniscourt



(b) Multi Labels from [25]:
Baresoil, Buildings,
Cars, Court, Grass,
Pavement, Trees

Figure 1.3: Example image from the UC Merced dataset shown with single [24] and multi-label [25] annotations.

With the growing annotated benchmark archives, deeper and complexer supervised state-of-the-art RS CBIR methods were introduced [1, 2, 25, 26, 28–30]. Although supervised methods dominate the leaderboard for RS benchmarks, they require labeled archives. Even if recent RS archives cover multiple countries [26], there is no guarantee that the model generalizes to other locations. For satellite images, which produce Terabytes of data each day, the data acquisition is not the main hindrance but the annotation process. Labeling the data is labor- and

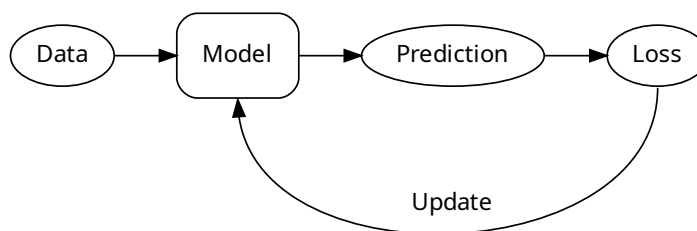


Figure 1.4: Unsupervised Learning Diagram

cost-intensive and is currently recognized as one of the main challenges in the general machine learning community [31, 32].

Semi-supervised techniques only require a fraction of the labels compared to classic supervised methods [33] but are restricted to the classes from the labeled subset. If some classes of the archive are unlabeled, manual intervention is required [3]. One method of learning with fewer labels is to train on *weak* labels first and then finetune on the labeled archive [34]. These weak, or noisy, labels are generated from unreliable sources, such as image titles or hashtags commonly used on social media [35]. There are other variants, but most semi-supervised methods require all relevant classes to be present in the true label set.

Unsupervised methods do not rely on any labeled data, neither for training nor for finetuning on different archives. Some unsupervised methods are data-independent and do not define a loss to update the model. Others *learn* from the data and iteratively update the model, as shown in Fig. 1.4. As RS image databases are one of the fastest-growing archives, it is crucial to minimize the manual work and to investigate unsupervised methods for the RS domain. There are various types of unsupervised methods, with contrastive self-supervised representation learning becoming one of the most popular methods in the classic computer vision domain [36–45]. Although some of these methods have been used and extended to the remote sensing domain, little work has been done to widen the scope of contrastive self-supervised representation learning to content-based image retrieval for remote sensing imagery.

The thesis will investigate if these contrastive self-supervised methods are a promising unsupervised CBIR training procedure for remote sensing images. Due to significant differences between the natural and remote sensing domains, a systematic re-evaluation of common suggestions — such as the augmentation pipeline — is necessary. Furthermore, a method-independent metadata-guided sampling framework is introduced. The proposed sampling framework is used to test assumptions about the interaction among samples in batches of the underlying contrastive method and improve overall retrieval performance.

The thesis is structured as follows: The upcoming Chapter 2 generally introduces self-supervised representation learning and motivates contrastive self-supervised learning as the relevant learning strategy. From the ever-growing field of contrastive self-supervised learning, three state-of-the-art methods (SimCLR, Barlow Twins, and BYOL) from the natural image domain are presented in detail. Chapter 3 motivates the necessity to re-evaluate the standard augmentation pipeline for remote sensing data. Based on the unique properties of remote imagery, a new augmentation pipeline is proposed. Inspired by recent work in the RS domain, Chapter 4 motivates a novel metadata-based sampling framework and highlights the possible use-cases and limitations. The sampling framework is tailored explicitly for contrastive self-supervised representation learning methods applied to the remote sensing domain. Chapter 5 presents the underlying dataset for all experiments and the specific experimental setup. The experimental results are covered in Chapter 6. Finally, Chapter 7 concludes the thesis and presents possible future work.

2 Self-Supervised Representation Learning (SSRL)

Designing tasks that result in non-trivial models without requiring any labels is the key challenge of unsupervised learning. The resulting model should detect and differentiate objects from each other without being explicitly taught what these objects are. Many different research directions try to accomplish this feat. One of the most popular approaches is the self-supervised representation learning (SSRL) training regime.

The following chapter will give a broad overview of the different self-supervised learning strategies and then motivate and focus on the best-performing variant, contrastive self-supervised learning. Three state-of-the-art contrastive self-supervised learning methods will be presented in detail, as these will be adapted to the RS domain and optimized for CBIR.

2.1 Introduction to SSRL

Instead of relying on labor-intensive human-annotated or noisy labels, self-supervised methods generate *pseudolabels*. These pseudolabels are the result of a

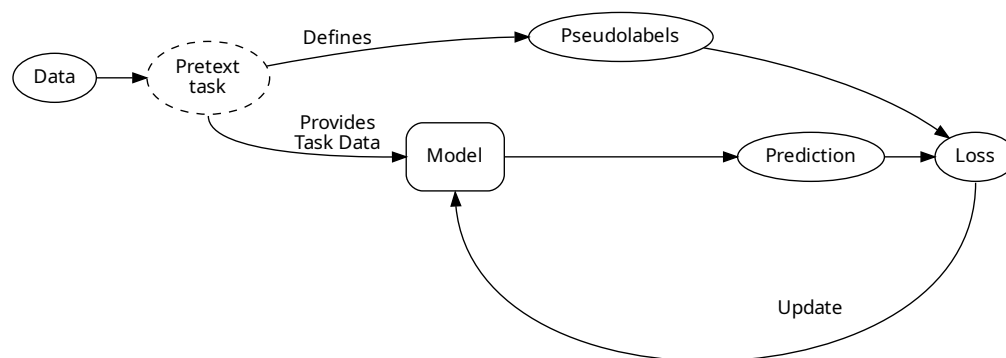


Figure 2.1: Self-Supervised Representation Learning Diagram: The pretext task defines how the pseudolabels are generated and may change the input data of the model.

pretext task. The pretext task is a pre-designed challenge for the network that is based on the data itself. By solving the pretext task, the model learns relevant visual representations or features. Fig. 2.1 visualizes the training loop of a general self-supervised method. The specific pretext tasks are very different from each other and depend on the specific self-supervised method. For images, Jing and Tian [46] defined the following three categories of self-supervised methods:

Generation-based methods: The model has to *generate* image data to match the pseudolabel. For example, the pretext task could be to colorize a grayscale version of the input image. The corresponding pseudolabel would be the original colored image. The model would have to predict the color data of the original image given the grayscale image.

Context-based methods: These methods define a pretext task with context features derived from the input image itself or context similarity. The pretext task could be to *undo* a randomly applied rotation to an input image. In this scenario, the context is the spatial structure of the image. The pseudolabel would be the exact amount by which the image was rotated.

Free semantic label-based methods: These methods bypass the manual annotation effort by automatically generating *ground truth* labels. Tools that create synthetic objects or scenes, such as game engines, are used to generate labels for *free*. The training loop would be more similar to a supervised variant with the main difference that the labels, or pseudolabels, were automatically generated and that the input images are synthetic.

With the provided categorization, the general pros and cons of each self-supervised method type can be reviewed: Game engines are able to produce *true* labels for realistic-looking images [47], but the domain gap between the synthetic and real-world images causes generalization issues [46]. Free semantic label-based methods are well-fit for domains where the data acquisition or verification is the bottleneck. For fields where the actual annotation process is the prohibiting factor, the other pretext tasks are preferred.

Popular generation-based self-supervised methods require particular architectures [48, 49]. These architectures are designed to solve the pretext task and be *directly* applied to other images after training [46]. If the pretext task is to inpaint parts of an image, the trained model will probably be used for the same task. Usually, the goal is to optimize the image generation-task, and not to learn generalizable representations for other tasks. Although some work has been done to learn generalizable representation through generation-based methods [50, 51], they are outperformed by context-based methods.

Context-based methods have recently gained popularity due to their superior performance compared to other self-supervised methods and their unrestrictiveness regarding the underlying architecture. The *supervisory* signal of these pretext tasks utilizes context similarity, spatial, or temporal structure.

The first well-performing context-based methods used the spatial structure to design pretext tasks, such as solving jigsaw puzzles [36, 52] or undoing random rotations [39]. In order to solve these spatial structure tasks, the model has to inherently learn visual features to determine the *correct* arrangement or orientation of objects.

During the design of pretext tasks, extra care must be taken to ensure that the task is not too easy or too difficult. If the task is too ambiguous, the model will have issues converging and learning valuable features [52]. The model may also learn *unusable* features if a trivial solution that bypasses the original pretext task can be learned [36].

The pretext task always has to be reviewed in the context of the specific application domain. For example, due to the rotation-invariance and the high spatial resolution of remote sensing imagery, undoing image rotations or solving jigsaw puzzles are ill-posed problems as RS pretext tasks. Compared to spatial structure tasks, the context similarity methods are less restrictive regarding their application domain.

The general idea of context similarity methods is to cluster the data into *similar* groups. Each element of a group should be similar to all the other elements and dissimilar to other groups [46]. The context similarity can be formulated as a contrastive or predictive task. The latter tries to predict to which *group* an input belongs, with the implication that these groups have to be generated in an unsupervised manner [37, 38]. The need to explicitly group the data is one of the reasons why contrastive pretext tasks are preferred. As the contrastive tasks do not explicitly group the data, they are less restrictive regarding the design and can usually be trained in an end-to-end fashion. Currently, contrastive self-supervised methods dominate most unsupervised benchmarks.

Due to the leading performance of contrastive self-supervised methods in the natural imagery domain, this self-supervised learning regime will be the main focus point of the thesis. Another crucial reason is that learning by comparing and grouping data should translate well to CBIR tasks.

2.2 Contrastive SSRL (CSSRL)

The thesis will focus on contrastive self-supervised representation learning (CSSRL) methods since they dominate various self-supervised learning benchmarks and seem well fit for CBIR. The following section will formally introduce CSSRL and provide a general framework to compare current state-of-the-art methods against each other. Three CSSRL methods (SimCLR, Barlow Twins, and BYOL) will be presented in detail, as these will be adopted to the RS domain and evaluated based on their CBIR performance. They will also be used as baselines for the proposed metadata-guided sampling procedures.

Contrastive tasks learn by applying a comparison among the input samples. The

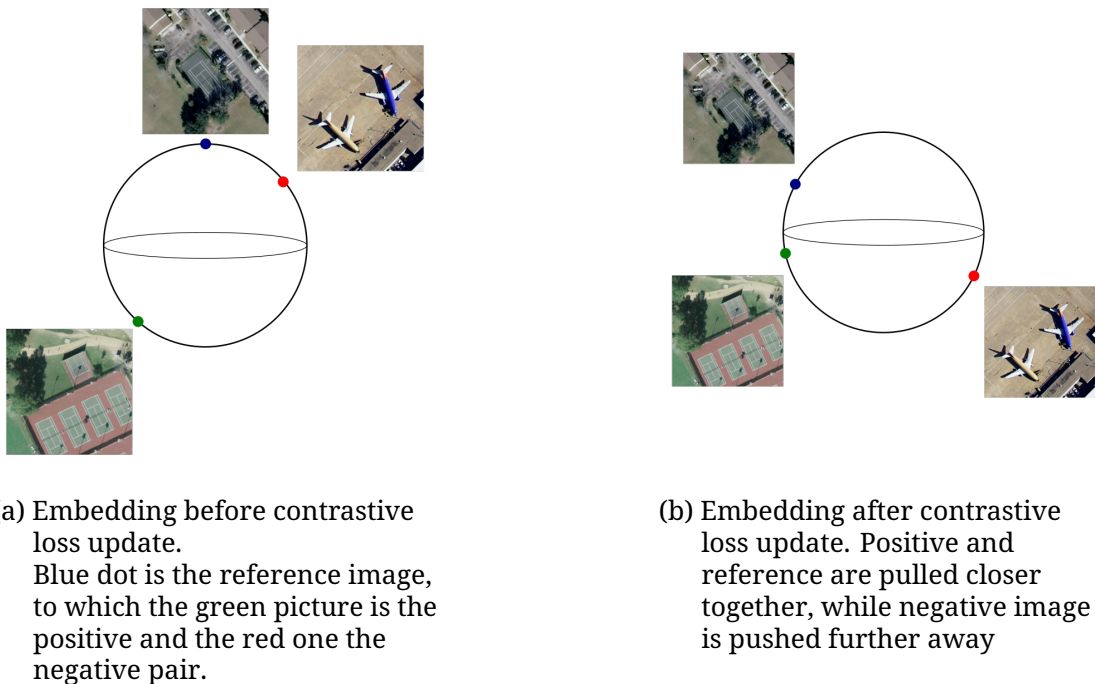


Figure 2.2: Visual contrastive loss update example on a sphere.

key idea is to maximize the similarity between samples from the same group and minimize the similarity to other groups. The similar elements are also referred to as *positives* and dissimilar ones as *negatives*. An example contrastive learning step is shown in Fig. 2.2. Here, the similar samples are pulled closer together, and the dissimilar ones are pushed further apart in a spherical embedding space.

For every input $\mathbf{x} \in \mathbf{X}$, where $\mathbf{X} := \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_A\}$ and A defines the dataset size, a positive distribution $p^+(\cdot|\mathbf{x})$ and a negative distribution $p^-(\cdot|\mathbf{x})$ can be defined. Given an input \mathbf{x} sampling from these distributions yield positive $\mathbf{x}^+ \sim p^+(\cdot|\mathbf{x})$ and negative $\mathbf{x}^- \sim p^-(\cdot|\mathbf{x})$ pairs. Usually, a contrastive method will contrast an input against both positive and negative pairs.

How the similarity distribution is generated depends solely on the specific contrastive method. Supervised contrastive methods may use the class-level information to define positive and negative pairs [53]. Images from the same class would be defined as positives and from other classes as negatives.

As in the relevant self-supervised scenario, no label information is assumed; positive and negative pairs have to be selected differently. The predominant method is to *generate* positive pairs through augmentation and *defining* the remaining images as negatives [40–42]. Fig. 2.3 shows how an input image may be augmented into different views to generate positive pairs. Fig. 2.3c is the positive pair to Fig. 2.3b and vice versa. Note that some methods do not define a negative distri-

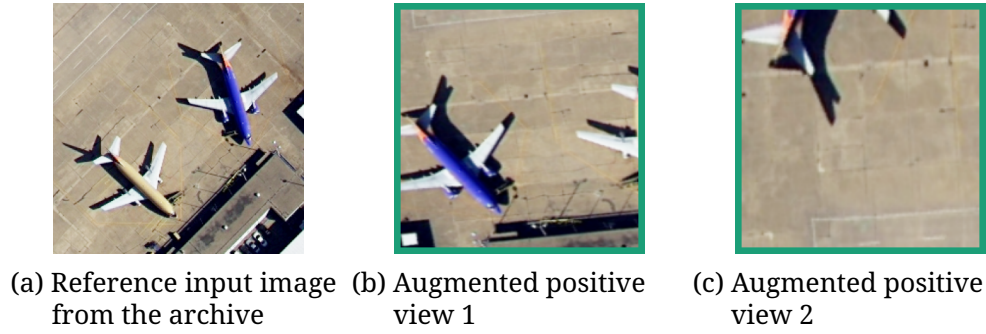


Figure 2.3: Visualization of positive view generation through image augmentation.

bution and only contrast among positive pairs (only pull similar images closer together).

The data has to be passed through a model to generate the *contrastive* embeddings. Fig. 2.4 visualizes the steps through a general contrastive self-supervised framework. As previously mentioned, the input images are augmented first. The specific augmentation operator t is sampled from an augmentation set T for each positive pair of an input \mathbf{x} . The augmented views $\tilde{\mathbf{x}}_*$ are passed through an encoder network.

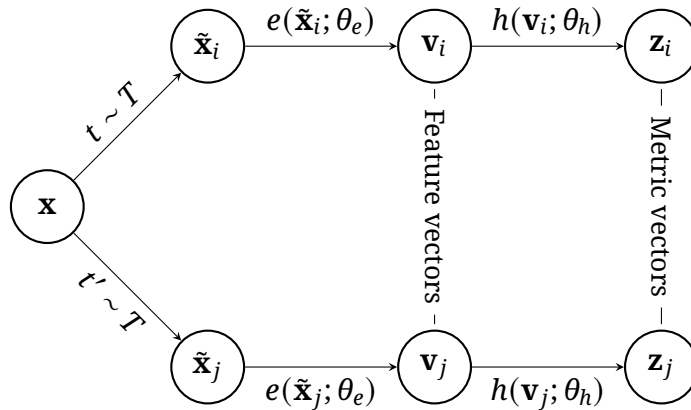


Figure 2.4: Steps through a general contrastive self-supervised framework.

Formally, the non-linear encoder $e(\mathbf{x}; \theta_e)$ converts an high-dimensional input image with C channels, a height of H and width of W ($\mathbf{x} \in \mathbb{R}^{C,H,W}$) into a lower-dimensional representation vector $\mathbf{v} \in \mathbb{R}^d$ with the learnable parameter set θ_e . A vector \mathbf{v} in the feature space \mathbb{R}^d is an abstraction of an input image to its essential properties. These representational feature vectors are used for downstream tasks, such as content-based image retrieval.

The feature vector \mathbf{v} passes through a projection head $h(\mathbf{v}; \theta_h)$ given by the

parameter set θ_h , transforming the input \mathbf{v} into a metric embedding $\mathbf{z} \in \mathbb{R}^{d'}$. The metric embeddings \mathbf{z} are combined with the group information in the contrastive loss function to maximize the similarity between positive pairs.

The contrastive loss enforces high similarity among the features in metric space by *teaching* the model to become invariant to *irrelevant* differences among the positive pairs [13]. In the self-supervised scenario, *irrelevant* refers to the applied data augmentation techniques.

The invariance to the applied augmentation techniques motivates the necessity of a separate projection head h . The projection head may remove unnecessary information such as color or orientation information of an encoded object, as the metric embeddings should not be affected by such transforms [41]. However, these properties are often relevant for downstream tasks such as classification or segmentation tasks. Instead of using both networks, the projection head h is dropped after the self-supervised training, and only the encoder e is used for downstream tasks.

In the remaining chapter, the relevant state-of-the-art methods are presented. These contrastive self-supervised methods form the foundation of the following thesis. Firstly, these methods will be adapted to the RS domain and evaluated by their CBIR performance. Secondly, they will be used as a baseline for the proposed metadata-guided sampling strategies.

SimCLR

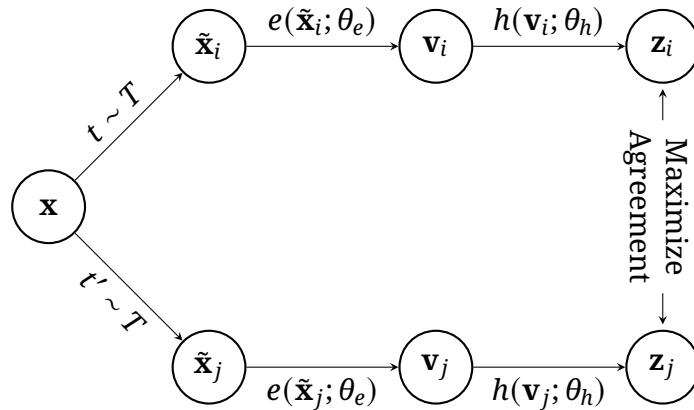


Figure 2.5: SimCLR Framework [41]

Chen et al. [41] developed a simple framework for contrastive learning of visual representations (SimCLR). A simple, yet competitive, contrastive self-supervised method. The SimCLR framework is presented in Fig. 2.5. The framework generates two augmented views for each input image (the positive pair) and defines the remaining images of the batch as negative pairs. As a result, the number of

negatives is a lot larger than the number of positives.

The augmented images pass through the encoder and projection head networks to generate the metric embeddings \mathbf{z}_i and \mathbf{z}_j for each input \mathbf{x} of the batch with size b . The positive metric embeddings are pulled closer together, while the negative pairs are pushed further away from each reference image. To calculate the agreement between two embedding pairs, a variant of the noise contrastive estimate (NCE) [54] is used:

$$\ell_{\text{NCE}}(i, j) = -\log \frac{\exp(S(i, j))}{\sum_{k=1}^{2b} \mathbb{1}_{k \neq i} \exp(S(i, k))}, \quad (2.1)$$

where $S(\cdot)$ is a similarity measure function. SimCLR uses the normalized-temperature cross-entropy (NT-Xent) variant of the NCE loss, where the similarity function is defined as the cosine-similarity divided by a scalar temperature constant τ [41]:

$$S_{\text{NT-Xent}}(q, k) = \frac{\mathbf{z}^T \mathbf{k}}{\|\mathbf{z}\| \cdot \|\mathbf{k}\|} \cdot \frac{1}{\tau}. \quad (2.2)$$

Intuitively, the loss minimizes the distance between the normalized metric embeddings that stem from the same input image and maximize the distance to all the other images in the current batch.

The main contributions of the work from Chen et al. [41] was the discovery of the importance of

- a non-linear projection head
- large batch sizes
- long training time
- a complex data-augmentation pipeline

Specifically, the requirement of strong data augmentations motivated the need to re-evaluate different augmentation techniques for the RS domain.

Barlow Twins

A different recent state-of-the-art self-supervised method is the Barlow Twins framework from Zbontar et al. [44]. The proposed framework can be seen in Fig. 2.6. Structurally, the framework is quite similar to SimCLR. Two augmented views are generated for each input image of the sampled batch and passed through the same encoder and projection head.

However, instead of comparing the similarity between the generated metric embedding vectors directly, the empirical cross-correlation matrix between the entire embedding matrices is calculated and compared against the identity matrix. Minimizing the cross-correlation between the metric embedding matrices reduces the redundancy within each representation vector and makes the embeddings invariant to the applied distortions.

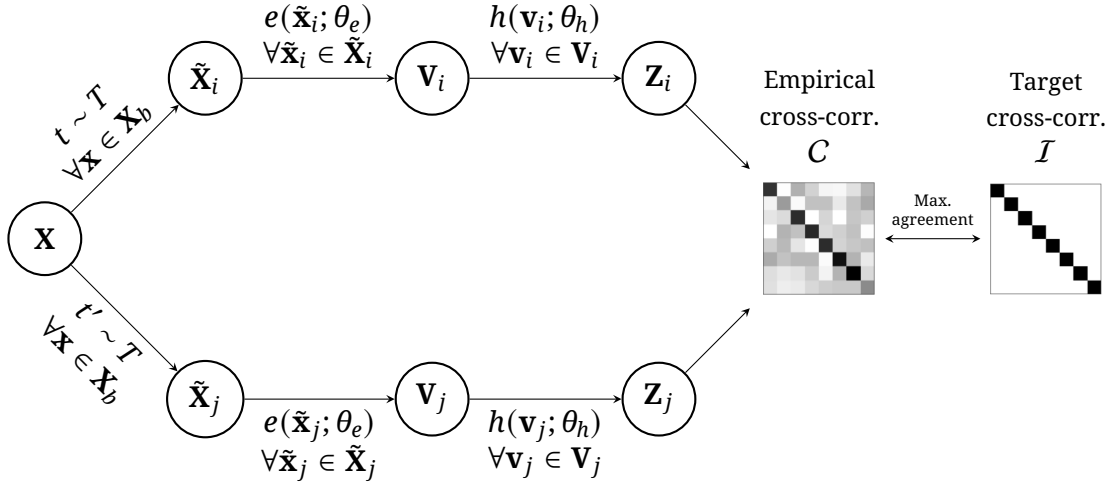


Figure 2.6: Barlow Twins Framework [44]

Formally, the following loss function is used:

$$\mathcal{L}_{\text{BT}} = \sum_k (1 - C_{kk})^2 + \lambda \sum_k \sum_{l \neq k} C_{kl}^2, \quad (2.3)$$

where λ is a positive scalar that weights the relevance of the second term compared to the first. \mathbf{C} is the empirical cross-correlation matrix computed between the embedding matrices \mathbf{Z}_i and \mathbf{Z}_j that have a shape of $b \times d'$, where b is the batch size and d' the metric vector dimension. Each entry of the empirical cross-correlation matrix is calculated with:

$$C_{kl} = \frac{\sum_b \mathbf{z}_{i,b,k} \mathbf{z}_{j,b,l}}{\sqrt{\sum_b (\mathbf{z}_{i,b,k})^2} \sqrt{\sum_b (\mathbf{z}_{j,b,l})^2}}, \quad (2.4)$$

where b indexes the sample from the batch and k and l index the component of the embedding matrix.

As the presented method does not compare the samples directly but the components of the embedding vectors, there is no notion of negative pairs. Still, the performance is heavily dependent on the applied augmentation techniques [44]. Note, even if there are no negative pairs, the samples in the batch itself are still relevant, as the vector components are contrasted against each other.

The main benefits of the Barlow Twins framework compared to SimCLR are the following:

- Works with smaller batch sizes
- Benefits from high-dimensional metric embeddings
- Higher scores in benchmarks than SimCLR

BYOL

The last investigated state-of-the-art CSSRL method is Bootstrap Your Own Latent (BYOL). The BYOL framework is visualized in Fig. 2.7. In contrast to the previous methods, BYOL does not use the same network for both branches. BYOL has an *online* and a *target* network that have different parameter sets, indicated by the index o and t respectively. The key idea is to train the online network to predict the target network's representation of the same input image under different augmented views.

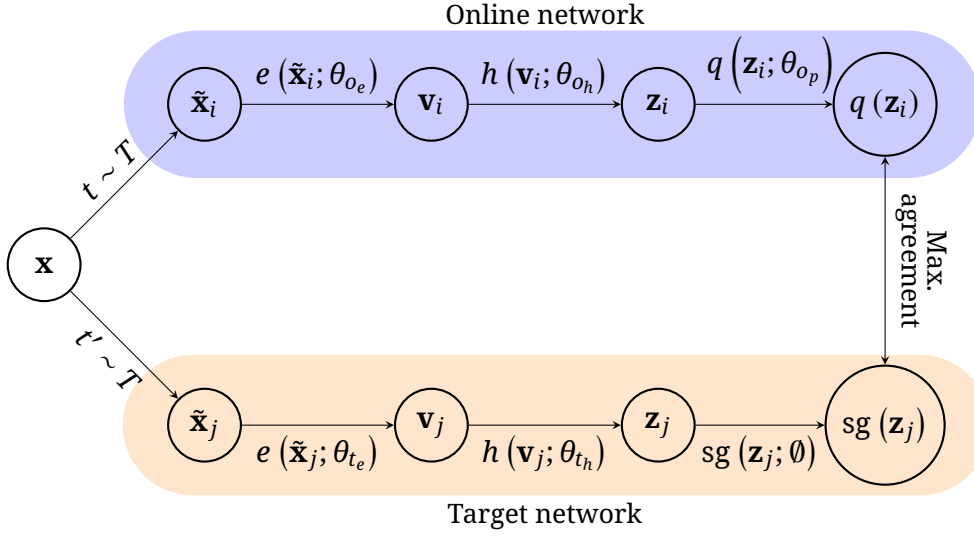


Figure 2.7: BYOL Framework [43]

The BYOL method only contrasts positive pairs and does not define negative pairs. The metric embeddings of the positive pairs are pulled closer together without being compared to other samples, making it robust against small batch sizes. An issue with this approach is that a trivial solution exists, where the representations of both networks collapse to identical embeddings. To avoid representational collapse, changes to the architecture were necessary. For one, a predictor q is added to the online network, and the target network is decoupled from the optimizer update step, indicated by the *stop-gradient* (sg) head.

The online network is updated directly through the loss function, while the target network is updated offline through a slow-moving average of the online network's parameters. For shorter notation, let $\theta_t := \theta_{t_e} \cup \theta_{t_h}$ and $\theta_o := \theta_{o_e} \cup \theta_{o_h}$. Concretely, the target network's parameters is updated with:

$$\theta_{t_{\text{new}}} \leftarrow \tau \theta_{t_{\text{old}}} + (1 - \tau) \theta_o, \quad (2.5)$$

where $\tau \in [0, 1]$ is the target decay rate.

The loss for the online network is calculated with:

$$\mathcal{L}_{\text{BYOL}} = \left\| \frac{q(\mathbf{z}_i)}{\|q(\mathbf{z}_i)\|_2} - \frac{\mathbf{z}_j}{\|\mathbf{z}_j\|_2} \right\|_2^2. \quad (2.6)$$

The main differences between BYOL and the previous CSSRL methods are:

- Higher scores in most benchmarks than SimCLR and Barlow Twins
- Less sensitive to batch size compared to SimCLR; more similar to Barlow Twins
- Only contrasts positive pairs; little interaction between images in a sampled batch
- Less sensitive to the specific data augmentation techniques used

All of these CSSRL methods share a similar augmentation pipeline that has been optimized for natural images. As the augmentation pipeline is a crucial component of contrastive self-supervised methods, the standard pipeline from the natural image domain has to be reviewed in light of the remote sensing domain. These methods will also be used as baselines for the proposed metadata-based sampling strategies.

3 Proposed Augmentation Pipeline for CSSRL in RS

Augmentation techniques are crucial for contrastive self-supervised representation learning methods. In all previously presented self-supervised methods, the data augmentation step is used to generate contrastive pairs given an input image.

Although there has been done some extensive research to find the most effective augmentation techniques [41, 43], the experiments were done on natural imagery. As remote sensing is a very different domain, the various augmentation techniques have to be re-evaluated. The re-evaluation is also necessary as multi-spectral imagery adds additional constraints to augmentation techniques.

In the remaining chapter, the following widely used augmentation techniques are presented:

- Dihedral Transformation
- Rotation
- Gaussian Blurring
- Resized Cropping
- Brightness Shifting
- Contrast Shifting
- Saturation and Hue Shifting

The exact transformation formulation, as well as visual examples, are provided. After reviewing these classic augmentation techniques for their use in the remote sensing domain, the chapter will conclude the most promising multi-spectral augmentation pipeline. Note that to be able to provide some example parameter values, the input images are assumed to be normalized to a range between 0 and 1.

Dihedral Transformation

The *Dihedral* transformation is composed of rotations and reflections. The dihedral group of images consists of four rotational and four reflection transformations. The rotational transformations of an input image are 0° , 90° , 180° , and 270° rotations. The reflection transformations are reflections along the y -axis, x -axis, lower-left to upper-right diagonal ($y = x$) and upper-left to lower-right diagonal ($y = -x$). Fig. 3.1 visualizes all Dihedral transformations given an input image.

The Dihedral transformation is quite interesting for aerial imagery, as these types of images are rotation-invariant. A natural image scene usually has an *upright* orientation, while bird's-eye view images do not. Also, the exact spectral

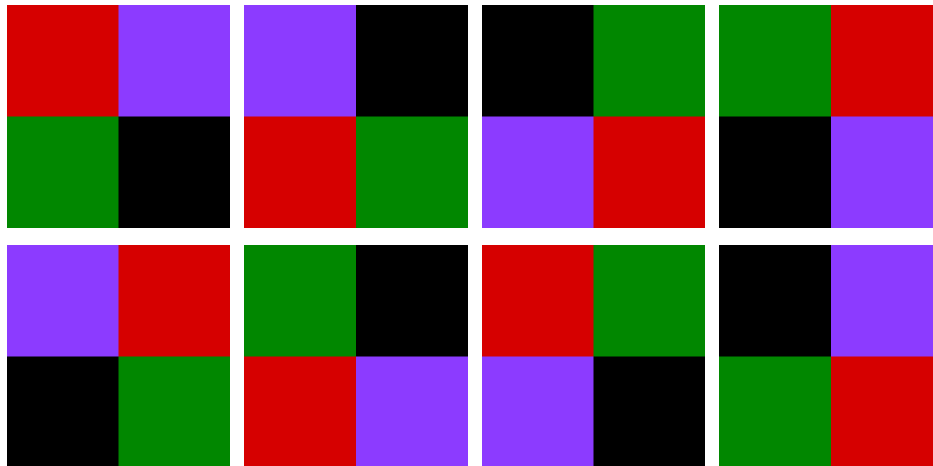


Figure 3.1: Examples of Dihedral transformations. First row shows the rotation (0° , 90° , 180° , and 270°) and the second row the reflection transformations (reflection along y -axis, x -axis, $y = x$, and $y = -x$ diagonal).

reflectance values of the sensed scenes are untouched, which might be more relevant for spectral-sensitive applications.

Positive pairs generated through rotation or reflection are semantically equivalent and *correct* (meaning the spectral reflectance values are untouched). As a result, the Dihedral transformation might be more valuable for contrastive self-supervised representation learning in the remote sensing domain, as suggested in the classic computer vision domain.

Rotating

The same argumentation applies to *continuous* rotation transforms. A Dihedral transformation only rotates in discrete steps (multiples of 90°). A continuous rotation transformation can rotate the image by any amount between $0^\circ - 360^\circ$. The main disadvantage is that it drops pixels near the corners and requires an interpolation step if the rotation angle is not a multiple of 90° . Applying rotations around 45° , 135° , 225° , and 315° to low-resolution satellite imagery introduces noise and potentially unwanted artifacts. Fig. 3.2 demonstrates how a continuous rotation drops information and adds noise to the reference image.

However, as rotation augmentation creates semantically identical positive pairs for aerial imagery, the continuous rotation transformation might be vital for the remote sensing domain. Carefully selecting the rotation values help to reduce the amount of information dropped near the edges of an image.

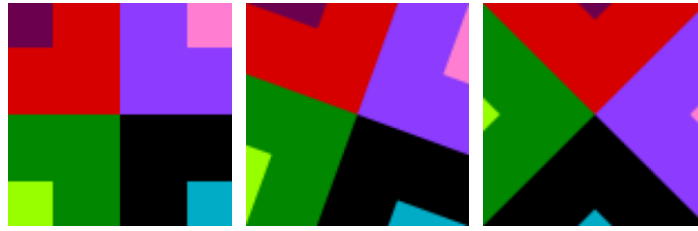


Figure 3.2: Examples of continuous rotations (0° , 20° , and 45°). Note that pixels near the edge are dropped and that the edge between the larger color-boxes becomes *blocky* in the rotated variants.

Gaussian Blurring

Gaussian blurring does not apply a geometric operation on the whole image but a convolution operation. The edges and colors of the input images are softened or *blurred* with a discrete Gaussian kernel, as shown in Fig. 3.3.



Figure 3.3: Example of Gaussian blurring with the input image on the left and the blurred image on the right.

Blurring has been shown to be helpful for contrastive self-supervised learning [41, 43]. The semantic content of positive pairs generated through Gaussian blurring is identical to the input image. The blurred pairs may help the model become less dependent on edge detection and spectral value matching, which is favorable for training generalizable models. The convolution operation modifies the underlying spectral reflectance values but does so by taking a weighted average of the local neighborhood. In conclusion, Gaussian blurring seems to be well-fit for aerial and natural imagery.

Resized Cropping

Resized cropping zooms into a rectangular region of the input image and resizes the crop back to the original dimensions. A couple of examples are visualized in Fig. 3.4. The default area of the zoom/crop step is randomly chosen to be between 8%–100% for each image [41, 43].

For natural imagery, even small crops tend to capture the relevant object or parts of it. For example, if the input is a picture of a dog, a small crop might still

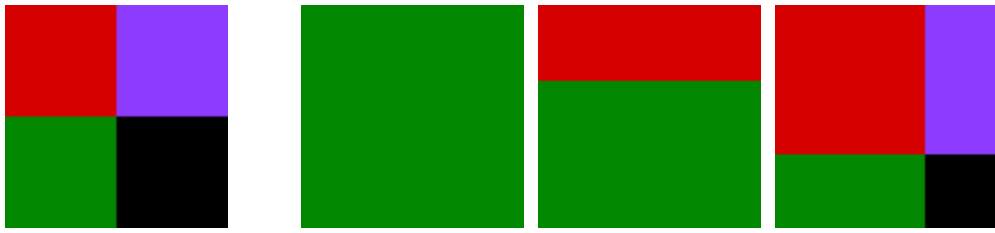


Figure 3.4: Examples of resized cropping given an input image (most left image).

be able to capture class-relevant features such as the face, ears, or paws. Remote sensing data is quite different in that regard. Due to the high spatial resolution of each pixel, crops are likely to drop entire classes from input images. Similar to how the first two crops of Fig. 3.4 drop the purple and black colors. Put differently, the likelihood of resized crops changing the semantic meaning of a remote image by dropping entire classes is comparatively high. However, low crops allow the model to become scale-invariant and to see more variability in the input. The higher variability may also counter possible *short-circuit* solutions, like histogram matching [41]. Especially since the natural image domain results strongly suggest using an aggressive resize crop augmentation strategy, the crop size will also be kept low for remote sensing images.

Brightness Shifting

The brightness corresponds to the amount of light in a scene. Adjusting the brightness can be formally described as adding a scalar $\beta \in [-1, 1]$ to all pixel values $p \in [0, 1]$ of an image. The scalar β is applied to all channels equally. The effect of different values for β can be seen in Fig. 3.5, where $\beta < 0$ decreases and $\beta > 0$ increases the brightness.

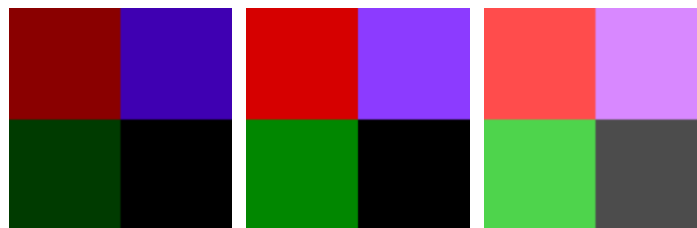


Figure 3.5: Examples of brightness shifting with $\beta = -0.3, 0$, and 0.3 .

The channels of multi-spectral images might have different value distributions. A value of 0.1 might be small for one channel and very large for the other. Therefore, globally changing the spectral reflectance values might hurt spectral-based object recognition, as it changes the semantic content of a multi-spectral image. Although brightness shifting is often applied in the natural image domain, the performance

of multi-spectral models might suffer from the augmentation technique. As a result, brightness shifting should be excluded from the default augmentation pipeline for multi-spectral images or only be applied with carefully selected values.

Contrast Shifting

Modifying the contrast of an image is equivalent to multiplying every pixel value with a scalar $\alpha \in \mathbb{R}^+$. Pixels are scaled towards the minimum ($\alpha < 1$) or maximum ($\alpha > 1$) values and away from the mean.

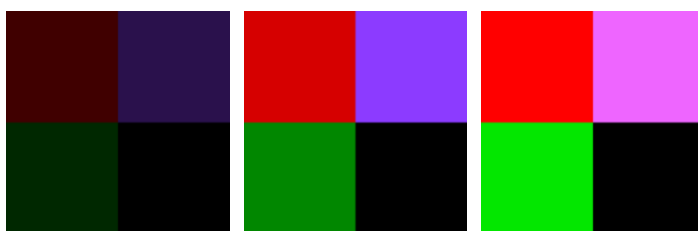


Figure 3.6: Examples of contrast shifting with $\alpha = 0.3, 1,$ and 1.7 .

For multi-spectral images, the issue is not that a specific α is multiplied to all channels, but that the spectral-reflectance values themselves can become so distorted that it hurts spectral-based recognition. Again, the semantic content of a multi-spectral image might be altered. As a result, contrast augmentation should only be applied with values close to 1 or be excluded from the default augmentation pipeline for multi-spectral images.

Saturation and Hue Shifting

Saturation controls the amount of *color* in an image. With zero saturation, a grayscale image is produced. Non-zero saturation values have no effect on neutral colors such as whites, grays, and blacks. A couple of example saturation transformations can be seen in Fig. 3.7.

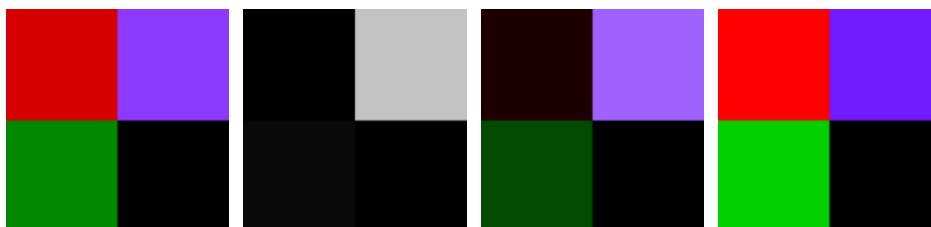


Figure 3.7: Examples of saturation shifting by setting the saturation to 1.0, 0.0, 0.7, and 1.4.

To modify the saturation the input image is first converted to hue, saturation, and value (HSV) space. In the HSV space one can also modify the hue of the image. A couple of hue transformation examples can be seen in Fig. 3.8.

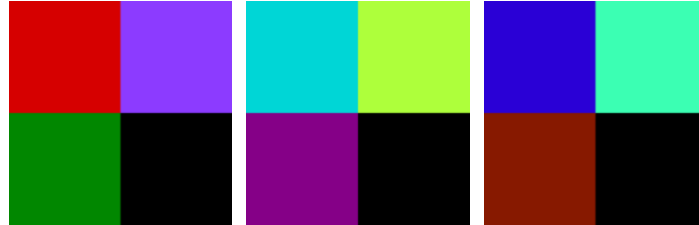


Figure 3.8: Examples of hue shifting.

The conversion to the HSV space is only defined for classic RGB images. As a result, the saturation and hue augmentation techniques cannot be applied to multi-spectral imagery. Note, that these augmentations are part of a popular augmentation composition called *color-jittering* [41]. Color-jittering combines brightness, contrast, hue, and saturation transforms into a single augmentation technique. For multi-spectral images, the color-jitter augmentation simplifies to a brightness and contrast transform.

Proposed Augmentation Pipeline

The special properties of remote sensing images require modifications to the default augmentation pipeline. The default augmentation pipeline recommended by Chen et al. [41] uses resized cropping, horizontal flipping, color-jittering (brightness, contrast, hue, and saturation shifting), and Gaussian blurring.

As previously mentioned, the hue and saturation shifting transformations are not well-defined for multi-spectral imagery. Due to the rotation invariance of the remote sensing data, the Dihedral and the rotation transformation should be added to the default augmentation pipeline. The main question is whether or not to include the remaining color transformations (brightness and contrast shifting). The issue is that color transformations might break the ability of the model to identify objects based on their spectral reflectivity. Conversely, Chen et al. [41] have shown that the color augmentation helps the model to not rely on histogram matching to find the positive pairs. Since the main benefit of multi-spectral imagery is the ability to distinguish objects based on their spectral reflectivity, the recommended pipeline does not include any color augmentation techniques. The experimental results section will quantitatively evaluate the decision to exclude color augmentations.

In summary, the following default augmentation pipeline for multi-spectral imagery is recommended: resized cropping, Dihedral transformation, rotation, as well as Gaussian blurring.

4 Proposed Metadata-Guided Sampling Framework for CSSRL in RS

Since contrastive self-supervised methods learn by *comparing*, the following chapter will motivate and showcase the benefit of a novel metadata-guided sampling framework. After introducing the general idea of utilizing metadata to guide the sampling process and relating it to previous work from the RS domain, the implications are discussed in detail.

4.1 Motivation

The key distinction among the CSSRL methods is what they compare and how they learn from the comparison. As discussed in the previous chapter, the positive pairs are usually generated through data augmentation techniques. An example is shown in Fig. 4.1. How these views are then contrasted depends on the CSSRL method.

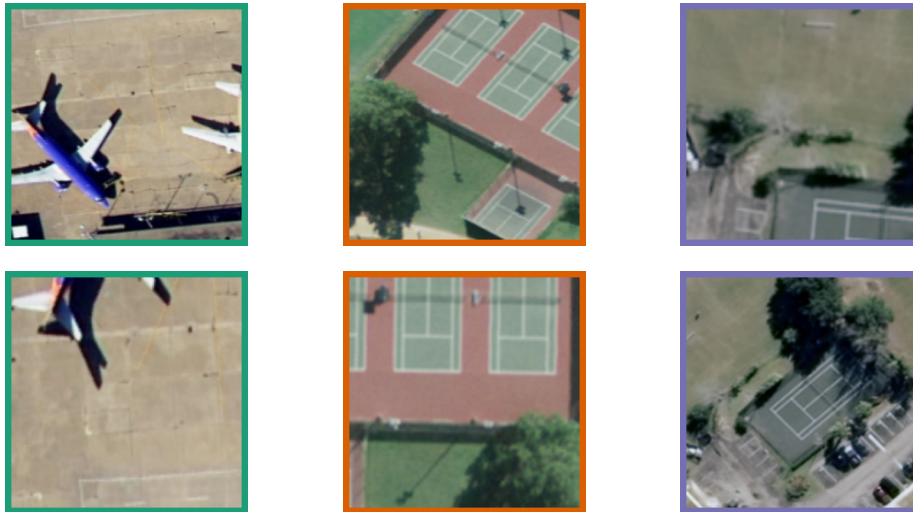


Figure 4.1: Positive pairs example, where same colored images are defined as positive pairs.

SimCLR generates a positive pair for each input image and defines the remaining images in the batch as negative pairs. The distance to the positive pair is minimized,

while the distances to the negatives are maximized. An issue with this approach is that a randomly sampled batch might have some *false-negatives*, from a label-based viewpoint. For example, in Fig. 4.1 there are two “tennis court” images. The *false-negatives* could reduce the overall performance or slow down the training since the model is effectively learning *wrong* relations. Conversely, if the samples in the current batch are too easy to distinguish, based on their color histogram, for example, it might trivialize the learning step.

Since most contrastive methods relate all samples in a batch against each other, the sampling procedure might play a crucial role in the overall performance. Work from the supervised contrastive domain supports the assumption, as research has shown that *smart* sampling improves performance [55] and reduces the training time [56]. Although it might be tempting to generate the hardest possible batches, hoping the model converges faster or learns fine-grained features, too hard batches might hurt the overall performance. Schroff et al. [53] have shown that it is crucial to choose the correct *hardness* level in order to maximize the benefit of contrastive learning.

The main obstacle of developing a smart sampling procedure for CSSRL methods is that no label information is available. Ideally, the sampling procedure should not impose any restrictions on the underlying architecture and be independent of the contrastive self-supervised learning strategy. The sampling framework should be able to assemble batches with varying difficulty levels. Furthermore, as the RS domain has almost unlimited access to data, a smart sampling framework should scale well with the dataset size and not add memory constraints.

Inspired by Tobler’s first law of geography [57], the upcoming sampling framework utilizes freely available metadata to guide the sampling procedure. Tobler’s law states that everything is related to everything else, but nearby things are more related than distant things. Most remote sensing imagery provides the exact location and acquisition time of the sensed image as metadata. This metadata can be used to cluster the data and then assemble batches by using different sampling strategies that control the *hardness*. Fig. 4.2 shows the general workflow.

By default, the framework utilizes location information to cluster the input data. According to Tobler’s law, the images within a cluster are more *similar* to each other compared to images from other clusters. Note that Tobler’s law is used as a *weak* signal and does not guarantee to group similar images together. Tobler’s law does not state that spatially close images share the same labels but could be interpreted to indicate that the labels within a cluster are less variable than compared to the complete set. Also, two patches might be similar, even if they do not share the same labels. They might share higher-order statistics like similar temperature or spectral-reflectance distributions. Note that the term *similar* heavily depends on the context.

After clustering the data, the only necessary modification to the training procedure is to decide on a sampling strategy. Assembling a batch from a single cluster (in-cluster sampling) translates to an approximately *hard* batch. *Easy* batches

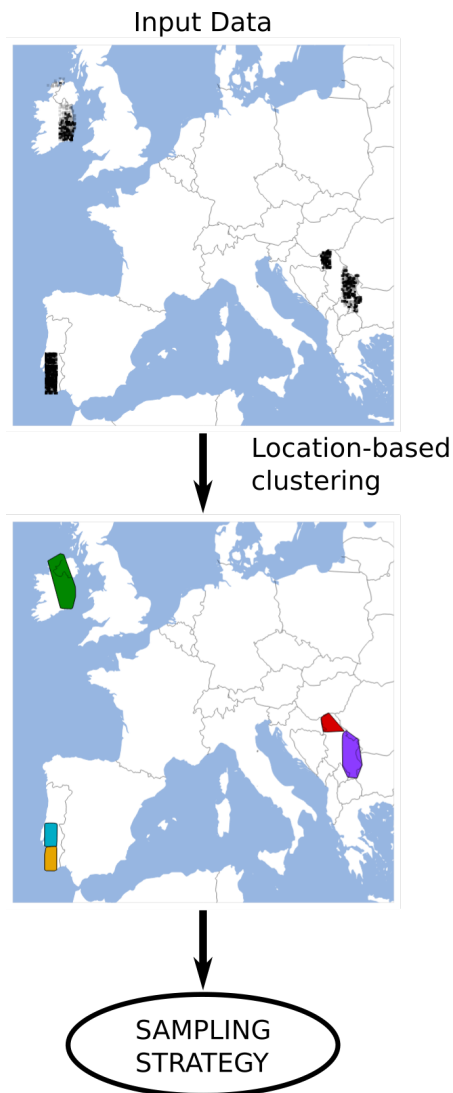


Figure 4.2: Location-Guided Sampling Framework: Given location-metadata, the input data is grouped into c clusters (here $c = 5$). These clusters are used as input for the sampling strategy.

could be generated by taking a single sample from each cluster (mixed-cluster sampling). Both of these approaches cause side effects that are thoroughly discussed in the following sections.

The remaining chapter will present the clustering and sampling strategies in detail after relating the framework to similar research in the remote-sensing domain.

4.2 Relation to Similar Work in RS Domain

The application of CSSRL frameworks in the RS community remains largely under-explored [58] but is gaining popularity. Remote sensing papers utilizing contrastive self-supervised learning strategies are becoming more popular due to advances in the natural image domain. Newer works integrate the unique properties of satellite imagery to improve performance further. However, the main focus point of these papers is to pretrain general models that are then finetuned for downstream tasks such as classification or object recognition [59–61]. In contrast to this work, they are not focused on the CBIR performance, nor are they focused on a general sampling strategy. Still, there is some notable overlap between the following papers and this thesis.

Jean et al. [59] implicitly use Tobler’s law to define positive pairs as image patches that are very close to each other and negatives as far away patches. This approach gives the specific location a very high relevance. Defining close patches as positive pairs may produce many false positives if the data source has a large spatial resolution. Generating positive pairs through image augmentation may be safer and allows the model to become invariant to unwanted distortions. The close patches are defined as patches that are within a specific radius of the source image. As a result, the method does not work well with sparse regions. Also, the proposed loss function is not easily adaptable to the current state-of-the-art CSSRL methods.

Kang et al. [60] further refined the idea by defining a new spatially augmented contrastive loss and a momentum update-based optimization. The momentum update-based optimization technique is based on a CSSRL framework from the natural image domain ([40]). Due to the strong connection to a specific CSSRL framework, the spatially augmented contrastive loss does not generalize well to other, possibly better, frameworks. The augmentation pipeline consists only of flipping, rotating, and the proposed spatial augmentation technique. The authors are not utilizing or analyzing other commonly used augmentation techniques at all.

Ayush et al. [61] have proposed a time-augmentation strategy and added a location-prediction task to an existing CSSRL method. The authors cluster the geographical data with k -Means [62] into c groups and let the model simultaneously learn to predict the source location of the input. The performance improvement of the location prediction task is comparatively low compared to the time-augmentation strategy. The authors limit the positive pair generation by only allowing temporal augmentation, which is very unusual for CSSRL methods. They make a strong assumption that the location is predictable from the images themselves, which might introduce a stronger bias compared to a solution where the location is only used as a weak signal. Also, their clustering strategy is ill-posed for a global-scale dataset, which is reviewed in detail in the following section.

In conclusion, the current thesis focuses on the application in the CBIR domain and proposes a method that is generally applicable to a wide variety of different

contrastive methods. The only requirement is that samples within a batch are contrasted against each other. Since the framework imposes minor restrictions on the underlying method, it can be quickly adapted to novel CSSRL methods. The framework is explicitly defined to work on global-scale datasets without negatively impacting the training time.

4.3 Location-Guided Clustering

The main motivation for the location-guided sampling strategies is Tobler’s Law. Tobler’s law states that nearby things are more related than distant things. After clustering the data based on the location metadata, the clusters themselves contain more similar samples than those from other clusters. Here, the location is used as a weak signal for the similarity. Weak, since sampling from a single cluster does not guarantee similar images and has no influence on the used loss function or how positive or negative pairs are defined. The only requirement for the underlying CSSRL method is that it must contrast images from different source images against each other. There are no other restrictions regarding the underlying architecture or method.

An unsupervised clustering algorithm is necessary to avoid human intervention and to train in an end-to-end fashion. By clustering the data and not defining a neighborhood radius, the preprocessing step is also well-defined for sparse regions. Since the clustered data is location metadata, it is easy to visualize the results. Although clustering large archives takes some time, the clustering step is done offline and only has to be done once per cluster configuration. Compared to the time spent training deep neural networks and searching for the best hyperparameters, the time spent clustering is negligible.

Ayush et al. [61] clustered the dataset with the k -Means [62] algorithm. In short, the k -Means algorithm works as follows:

1. Select k random *means* (usually random data points)
2. Assign each data point to the closest mean
3. Calculate the new mean for each cluster
4. Repeat the previous two steps until the assignments stop changing

Although the algorithm is popular and easy to understand, the application on geographical data is ill-posed. Using k -Means on data that is defined by the long- and latitude coordinates implies that the earth is projected onto a map. The original view against a projected view can be seen in Fig. 4.3. The projection allows working with euclidean space-based algorithms, such as k -Means, but introduces a couple of issues.

A result of projecting the earth *into* the euclidean space is that the spherical properties are lost. Algorithms cannot be aware of the wrap-around at the longitudes -180° and 180° (Fig. 4.3b). Also, if the value of only one axis changes, the actual distance heavily depends on the other value. An extreme case is if there

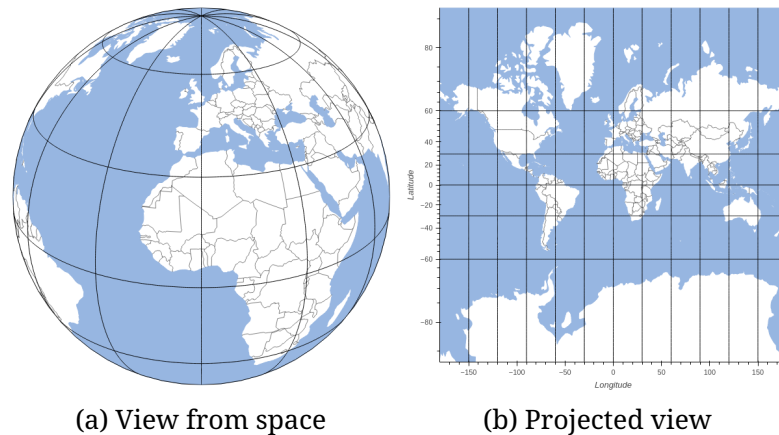


Figure 4.3: View on earth from space vs. projected map.

are points near the poles and equator. Let point one be $p_1 = (0^\circ N, 0^\circ W)$ and point two $p_2 = (0^\circ N, 1^\circ W)$. The actual distance is around 110 km. For two different points with the same long- but different latitude values: $p_1^* = (80^\circ N, 0^\circ W)$ and $p_2^* = (80^\circ N, 1^\circ W)$ the distance shrinks to about 20 km. The significant difference between the euclidean distance in the projected space and the actual distance could heavily distort the cluster results.

Applying k -Means clustering to small geographical regions can provide valuable insights into the data but is ill-fit for global scale clustering. Especially in the remote sensing domain, with an abundance of available data, a clustering algorithm should be chosen that will not distort the cluster results or introduce biases.

k -Medoids [63] allows to cluster data in euclidean and non-euclidean spaces, such as the haversine space (*spherical space*) for example. Instead of calculating the mean and *generating data* for each cluster, as k -Means does, the k -Medoids algorithm selects the most centrally located point (medoid). An issue with this algorithm is that it has a non-deterministic polynomial-time hardness and requires heuristic approaches to apply the algorithm to real-world datasets. Still, recent advances in research have considerably improved the performance of the heuristic approaches and allow k -Medoids to be used on global-scale datasets [64, 65]. Specifically, the latest publication from Schubert and Rousseeuw [65] will be used as the basis for the location-based clustering step. The algorithm will use the haversine space to *accurately* calculate the distance between satellite images. Note that k -Medoids is not restricted to the haversine space, and other data such as the label information could also be used to cluster the data in hamming space.

To give the number of clusters a more meaningful abbreviation, the thesis will use the variable c to define the number of clusters and not k . After generating the c clusters, the next step is to sample from these. The choice of c has one simple but essential implication. As the number of clusters c grows, fewer data points are assigned to each cluster individually. Depending on the sampling strategy, the

specific value of c might constrain possible sampling parameters.

To provide some real-world values, Table 4.1 shows some statistics about the clustering strategy applied to the BigEarthNet-S2-Summer archive. The statistics highlight that the difference between the smallest and largest cluster is quite large and that it heavily depends on the number of clusters c . After clustering the data, the images can be sampled in a *smart* way. Note that the dataset will be introduced in detail in the following chapter.

Table 4.1: Example Cluster Statistics: Given the number of clusters c , the minimum, maximum, mean, and the relative difference between the maximum and minimum value are displayed.

number of clusters c	min	max	mean	relative max/min difference
10	3470	8197	6559	2.4
16	1048	8197	4099	7.8
64	356	1813	1024	5.1
128	72	937	512	13.0
256	72	473	256	6.6
512	57	226	128	4.0

4.4 In-Cluster Sampling

One sampling strategy that can be applied after clustering the data is *in-cluster* sampling. Here, the current batch that is used to update the model only contains images from a single cluster. The in-cluster sampling strategy is visualized in Fig. 4.4. According to Tobler’s law, sampling from a single cluster should give more similar images and, therefore, create *harder* batches. If the contrastive task is too easy, then the benefits of creating harder batches would be a decreased time until convergence and higher final performance.

The hardness is be controlled over the area a cluster covers. Larger clusters would generate approximately easier batches, while smaller ones would generate harder batches. However, there are a couple of restrictions regarding the previous clustering step. The main issue is that the number of images per cluster should never subceed the batch size. Otherwise, images would be present multiple times in a batch, leading to an ill-posed pretext task. SimCLR, for example, defines all of the remaining images in the batch as a negative pair and the other augmented view as a positive pair. If the identical image is augmented multiple times, it is impossible to identify the correct pairs since there are now multiple solutions.

Since the k -Medoids algorithm does not partition the data into equally large clusters, particular focus should be put onto the smallest cluster. Note that clusters that are *too* small cannot be simply merged since they do not necessarily have

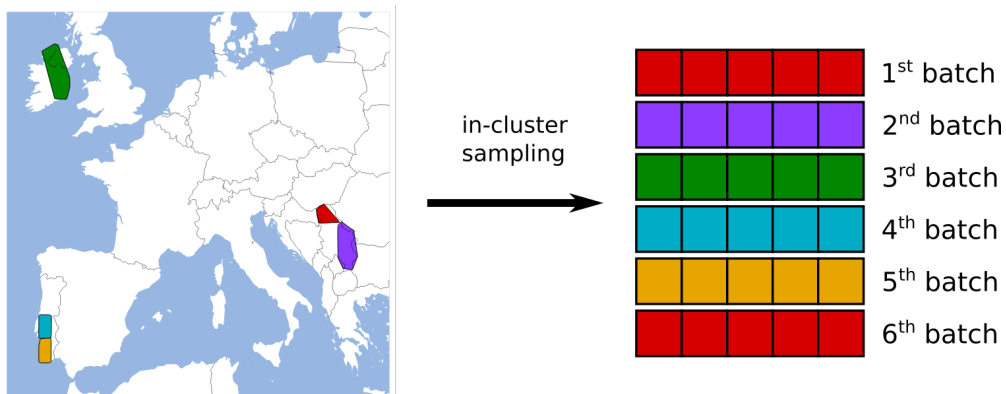


Figure 4.4: In-cluster sampling visualization. Given location-based clusters, only sample patches from one cluster per batch.

to be spatially close to each other. A simple solution would be to decrease the batch size, but depending on the underlying algorithm, it may hurt the overall performance [41]. Looking at the example values provided in Table 4.1, the largest possible batch size for $c = 16$ is only 1048. For comparison, the SimCLR paper recommends batch sizes starting around 2048.

A different side-effect of the in-cluster sampling strategy is that patches from smaller clusters are sampled more often compared to the larger clusters. The impact largely depends on the specific choice of c and the underlying dataset. The imbalance can be evaluated by looking at the cluster-size distribution. Note, an imbalance towards smaller clusters (sparser regions) may be beneficial for some use-cases as these *rarer* regions would be oversampled.

In conclusion, the in-cluster sampling strategy samples all *similar* images from a single cluster. The cluster size, which is indirectly controlled via the number of clusters, influences the approximate hardness and largest possible batch size. Note that the framework allows updating the clusters during training. An advanced sampling strategy could start with few but large clusters and gradually increase the hardness by increasing the number of clusters.

4.5 Mixed-Cluster Sampling

The complementary approach is *mixed-cluster* or *interleaved* sampling. Instead of creating the complete batch from a single cluster, the batch is assembled from multiple clusters. The mixed-cluster sampling strategy is visualized in Fig. 4.5 The idea is to generally produce *easy* batches, or batches with a large dissimilarity, by ensuring that each patch is *far* away from all the other patches. In contrast to the in-cluster sampling strategy, the mixed-cluster variant does not restrict the batch size. The number of clusters c can be simply defined as the batch size b .

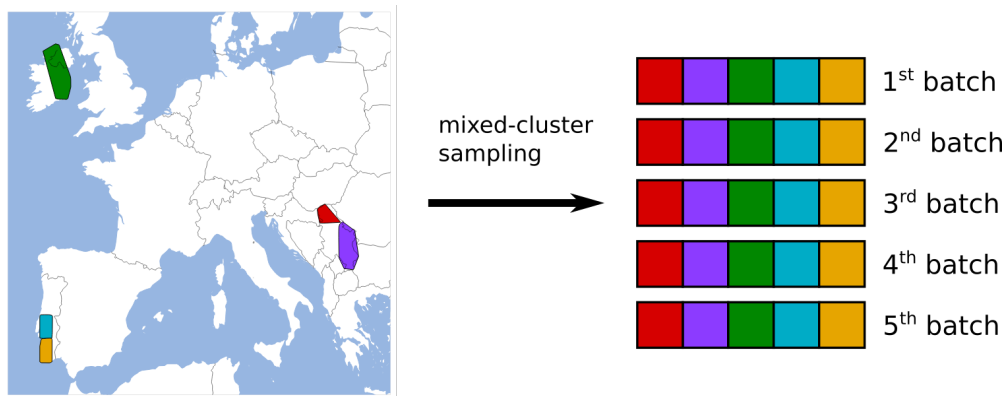


Figure 4.5: Mixed-cluster sampling visualization. Given location-based clusters, sample a single patch per cluster to assemble the batch.

Although, as previously discussed, many small clusters introduce a bias towards sparser/smaller regions. For large batch sizes, the smallest cluster could collapse to only a few images. In the extreme case, a cluster might only contain a single image. As a result, that specific image would be present in *all* batches. Even if the mixed-cluster strategy does not restrict the batch size, large batch sizes could lead to strongly oversampled small regions that could negatively impact the performance. Special care should be taken when large batch sizes are considered. Looking back at Table 4.1, the mixed-cluster strategy is well defined for $k = 512$, whereas the in-cluster sampling method would be limited to a batch size of 57, which is relatively low. Note that one could also sample multiple images per cluster, but that would hurt the primary goal of sampling patches that are all far away from each other.

The mixed-sampling strategy is well-fit for contrastive tasks that are too hard and require easier batches. The strategy is also favorable for self-supervised strategies that benefit from a larger variance within batches.

4.6 Key Points of Proposed Metadata-Guided Sampling Framework

The main motivation for the location-guided sampling strategies is Tobler’s Law. After clustering the data based on the location, two proposed sampling strategies can be utilized. The in-cluster sampling strategy is better suited to increase the hardness of the batches compared to the classic random sampling. The main issue with this approach is that the batch size may be strongly limited due to the number of available images in the smallest cluster. Although, the effect heavily depends on the specific dataset and the choice for c . In the setting of CSSRL methods, the batch size will probably be comparatively small. The mixed-cluster variant is designed

to create easier or more variant batches. The batch size is indirectly limited by the cluster size but is generally a lot larger compared to the in-cluster strategy.

In both cases, only the location metadata is used. The framework could easily be extended to include temporal information as well. The temporal information could be used to pre-split the data, further increasing the similarity among samples in the clusters. The different sampling variants could also be combined to create a complex sampling strategy. One could start with easy batches to train the model faster and then *finetune* the model by only creating hard batches at the end of the training.

Both variants use the same cluster results. The clusters only have to be calculated once for a specific number of clusters and are generated independently of the neural network training procedure. The effect on the overall training time is negligible. During training, there is no performance overhead. A general issue with the approach is that it assumes a large archive. The batch size will probably be too restricted on smaller datasets, and the methods could negatively impact performance if sparse regions are heavily oversampled. However, since there is an abundant amount of data in the RS domain and most CSSRL methods require large datasets to function properly, the dataset size should not be an issue.

Both strategies will be applied to all of the previously introduced CSSRL methods (SimCLR, Barlow Twins, and BYOL). The only requirement to benefit from the smart sampling procedure is that the views from different images have to be contrasted against each other. Since both SimCLR and the Barlow Twins method contrast different images against each other, they should benefit from the smart sampling methods. The sampling methods were mainly motivated for SimCLR, but it should also translate to the Barlow Twins method, even if the Twins contrast feature vector components and not image views per se. BYOL only explicitly contrasts positive pairs and should not perform better due to a different batch hardness. However, as the proposed sampling strategy also oversamples sparse regions, it might affect retrieval performance and is worth investigating. Before moving on to the experimental results, the underlying dataset and the general experimental setup are presented in detail.

5 Dataset and Experimental Setup

One of the largest remote sensing archives, BigEarthNet-S2, was chosen as the main dataset for the experimental analysis of the thesis. The following chapter will start with a general introduction of the archive, the typical applications of the various multi-spectral bands, and introduce the two class-nomenclatures. The chapter will continue to present four popular retrieval metrics and discuss their implicit biases. The chapter concludes with the specific experimental setup used, such as model architecture, hyperparameter values, and implementation details.

5.1 The BigEarthNet-S2 Archive

Sümbül et al. [26] released the first large-scale remote sensing BigEarthNet archive in 2019. The original version, BigEarthNet-S2, consists of sensed tiles from the Sentinel-2 satellite. With the Sentinel-2 satellite as a data source, thirteen spectral bands with different spatial resolutions were available, from which all but one band were included in the BigEarthNet-S2 archive. Fig. 5.1 visualizes the spectrum of each band grouped by their spatial resolution. The spectrums range from visible (380 nm – 700 nm), near-infrared (700 nm – 1100 nm) to short-wave infrared (1100 nm – 3000 nm) light.

The low-spatial resolution 60 m bands are designed for aerosol (B01), water-vapour (B09) and cirrus cloud (B10) detection. Sümbül et al. [26] not included band B10 as B10 provides no surface-level information [66].

The 20 m bands in the near-infrared range (B05, B06, B07, B8A) are vital for differentiating vegetation from other objects. The remaining 20 m bands in the short-wave infrared spectrum (B11, B12) are helpful for snow, ice, and cloud discrimination [9].

The high spatial resolution bands B04, B03, and B02 are the *classic* red, green, blue (RGB) channels. The last 10 m resolution band, B08, covers a broader spectrum than B08A but is less resistant against water vapor contaminating the spectral reflectance. As some applications suffer from possible contamination, the narrower, lower spatial resolution band B08A was added. To cover all use-cases, both bands were included, even if they overlap in the sensed spectrums [67]. The combination of these spectrums results in images with twelve channels, compared to the classic three RGB channels.

Due to the different spatial resolutions of the twelve bands, the pixel dimensions are not identical. Every patch covers a region of 1200 m × 1200 m. Given the spatial

5 Dataset and Experimental Setup

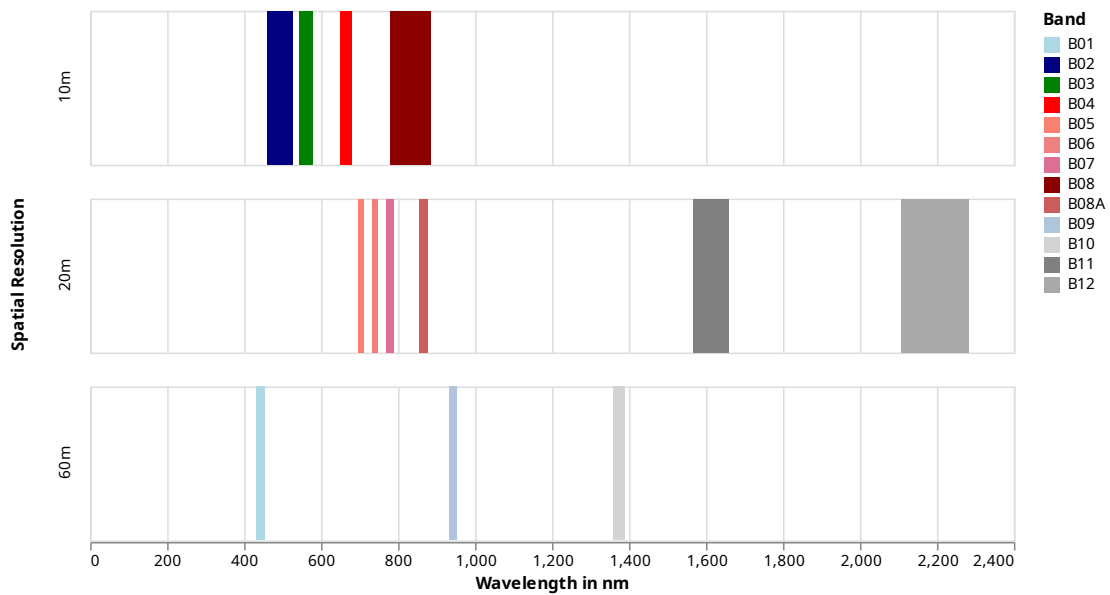


Figure 5.1: Spectral Bands from Sentinel-2 Satellite [9, Figure 3.5].

resolutions of 10 m, 20 m, and 60 m the respective width/height dimensions are 120 px, 60 px, and 20 px.

To visualize the sensed region, the bands can be visualized individually as shown in Figs. 5.2a to 5.2c, or the RGB bands can be combined to produce a true-color representation (Fig. 5.2d).

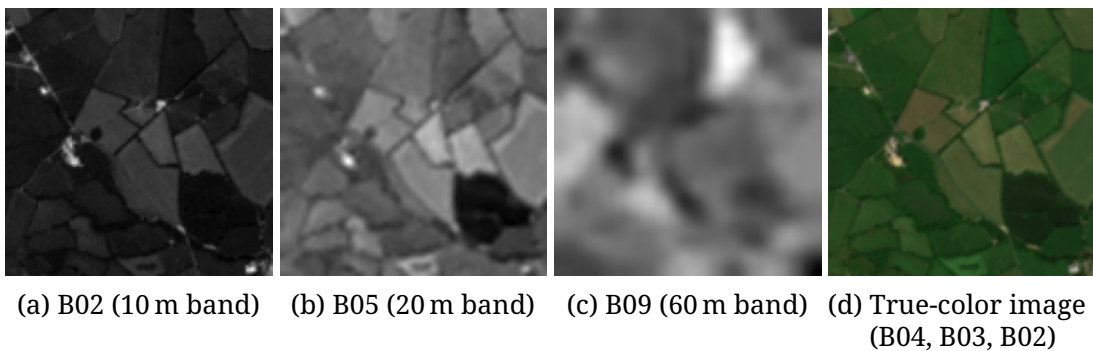


Figure 5.2: Example Patch Visualizations: Figs. 5.2a to 5.2c show individual bands interpolated to the same width/height; Fig. 5.2d combines the three RGB bands to produce a true-color image

BigEarthNet-S2 provides 590 326 of such patches from ten countries in Europe. Fig. 5.3 visualizes the exact regions of the sensed tiles. 61 707 of these patches are mostly covered by seasonal snow and 9280 by clouds.

Sümbül et al. [26] suggest removing these for scene classification and image

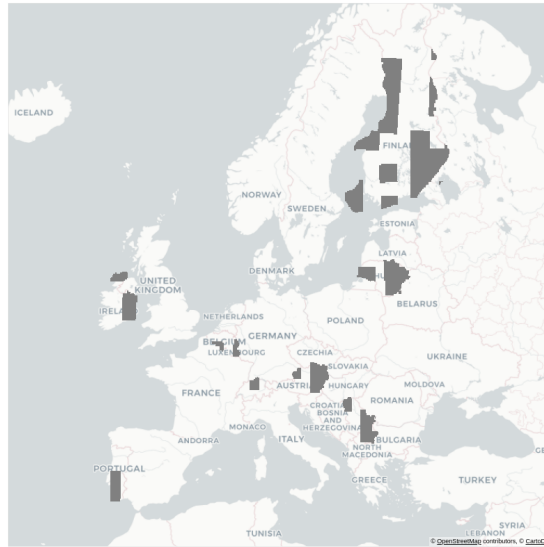


Figure 5.3: Distribution of BigEarthNet patches.

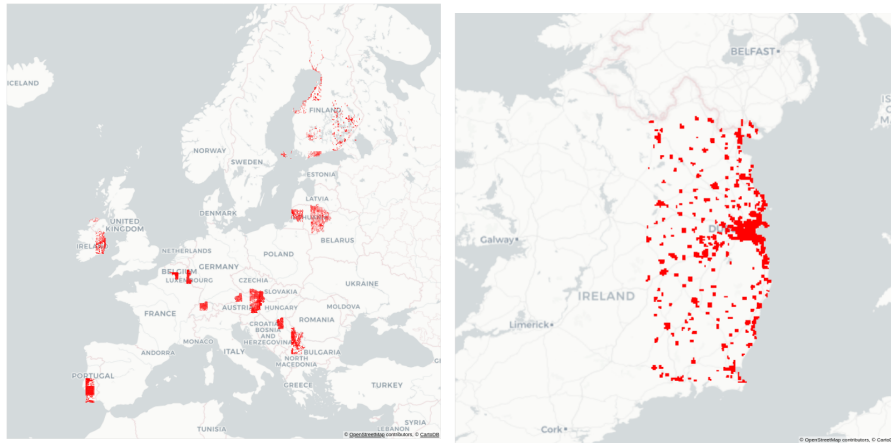
retrieval. Typical for remote sensing benchmarks, the BigEarthNet archive is a multi-label dataset. There are two proposed class-nomenclatures with either the original 43 or recommended 19 classes. The relation of both of these class-nomenclatures are shown in Table A.1. The recommended class-nomenclature drops 11 labels, resulting in 64 *classless* patches that have to be removed from the archive. Fig. A.1 shows example images that contains the respective class label for the 19 class-nomenclature.

Using the 19 class-nomenclature, one can plot all patches that contain a specific label on a map. Visualizing all patches that contain the “Agro-forestry areas” label, as done in Fig. 5.4a, highlights that the location metadata can provide valuable information about the possible content of a patch. For labels like “Urban fabric”, as shown in Fig. 5.4b, the value of utilizing the location metadata is less obvious. The “Urban fabric” patches seem evenly distributed, but compared to all patches (Fig. 5.3), there are excluded regions. Zooming closer into the image, as shown in Fig. 5.4c, one can see that the urban patches focus around specific regions. The visualizations shown in Fig. 5.4 support the use of the location information as a weak signal for similarity.

For the following experiments, the recommended 19 class-nomenclature is used. To reduce training time and to remove possible side-effects of seasonal changes, the default configuration will *only* use patches from the summer season. This specific subset will be referred to as BigEarthNet-Summer. The original training, validation, and test split is repurposed to define the training, query, and archive splits for the CBIR benchmark. The original training split is also used to train the model. The validation split is defined as the query and the test as the archive split. Each image of the query split will be used to retrieve a predefined number



(a) Visualization of patches containing “Agro-forestry areas”



(b) Visualization of patches containing “Urban areas”

(c) Visualization of patches containing “Urban areas”
(Zoomed into Ireland)

Figure 5.4: Visualization of BigEarthNet patches containing specific labels.

of samples from the archive. How the retrieval performance is quantified is described in the following section.

5.2 Evaluation Metrics

To quantitatively evaluate the image retrieval performance, the following widely-used image retrieval metrics for multi-label benchmarks were chosen [68, 69]:

- precision
- mean Average Precision (mAP)
- weighted mean Average Precision (wmAP)

- normalized discounted cumulative gains (NDCG)

The following paragraphs will show how these metrics are defined and highlight their pros and cons. After introducing the various metrics, the primary metric for the following experimental results evaluation is chosen.

Precision

In the general information retrieval domain, precision is defined as:

$$\text{precision} = \frac{\text{relevant retrieved results}}{\text{total number of retrieved results}} . \quad (5.1)$$

In the context of single-label image retrieval, a *relevant* result to an input/query image with the label l would be an image with the same label l . For example, if the input to the image retrieval system is an image with the label “Dog”. The system then returns two images, one with the label “Cat” and one with the label “Dog”. The precision score would be 0.5, as the returned images contain one relevant result from two.

For multi-label benchmarks, one has to define *relevant* first. Commonly, a result is relevant if the multi-label result shares *any* labels with the query image [68]. Formally, every multi-label image I has an associated label set L . A similarity score between two multi-label images can then be defined as the number of shared labels:

$$s(q, i) = |L_q \cap L_i| . \quad (5.2)$$

A relevant retrieval result would correspond to $s(q, i) > 0$.

In CBIR benchmarks, the scores for various numbers of retrieved images are reported and inspired the notation $\text{precision}@k$, where k defines the number of retrieved images from an archive set A . The precision score can then be formulated as:

$$\text{precision}@k(q) = \frac{\sum_{i=1}^k \mathbb{1}_{s(q,i)>0}}{k} , \quad (5.3)$$

where $\mathbb{1}_{s(q,i)>0}$ is an indicator function defined as:

$$\mathbb{1}_{s(q,i)>0} = \begin{cases} 1, & \text{if } s(q, i) > 0 \\ 0, & \text{otherwise.} \end{cases} \quad (5.4)$$

To better evaluate the performance of a retrieval system, the precision metric is usually averaged over all queries of a predefined query-set Q . The *overall* precision on a query archive Q is then defined as:

$$\text{overall precision}@k(Q) = \frac{1}{|Q|} \sum_{q=1}^{|Q|} \text{precision}@k(q) . \quad (5.5)$$

One of the main disadvantages of the precision metric in a multi-label scenario is the *relevance* definition. As only a single label must be shared among the query and retrieved image, the score might be ill-fit for complex imagery with many labels per image. A high precision score might hide underlying issues and not reflect human expectations.

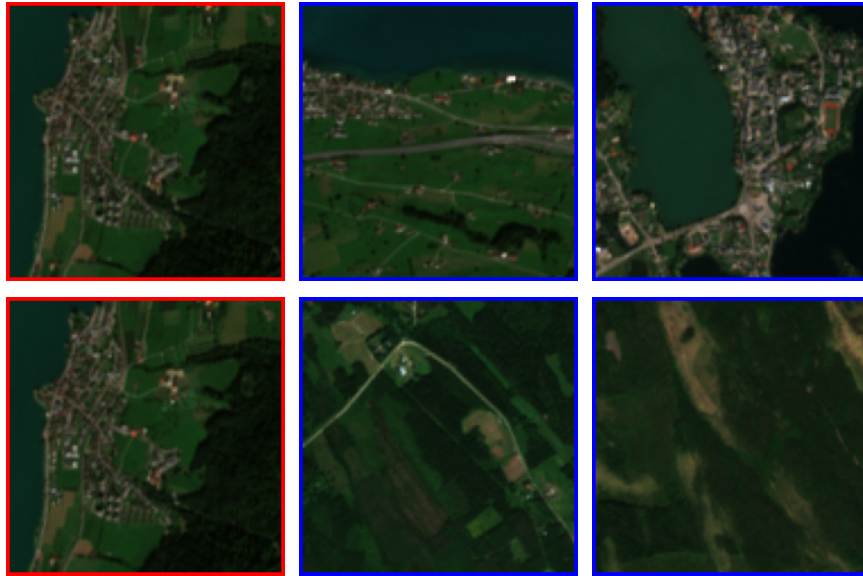


Figure 5.5: CBIR examples with precision = 1. The first row shows a retrieval system that matches all labels from the query image, the second row shows a system that only matches a single label from the query image.

See Fig. 5.5 for example: The red bordered query image contains the following labels: “Mixed forest”, “Complex cultivation patterns”, “Inland Waters”, and “Urban Fabric”. Both image retrieval systems have the same precision score of 1, as each retrieved image contains at least one overlapping label. The first system returned scenes with an identical label set, while the second system only retrieved patches that contain the “Mixed Forest” label. Quantitatively, both systems perform identical according to the precision score, while the first system performs qualitatively better.

An arguably different disadvantage is the invariance to the retrieval order. If the retrieval order should be considered relevant or not depends on the application. If the retrieval system is used for preprocessing or batching data and feeding it into a different order-invariant system, the retrieval order is not relevant. If, however, the system has a human interface, then the order might become crucial. A user might be inclined to not search through all retrieved images to find the best match, especially if k is large. The images at the end of the retrieval queue would become less likely to be examined [70].

The last disadvantage of the precision score is that k cannot become larger than

the smallest minority class. If there are only a few images of a specific class in the archive, the precision score might misrepresent the performance of the retrieval system.

Assume that there are only two images with a specific class l in the archive, but the archive contains thousands of other images. Setting $k = 10$ and using a query image with the class l , the precision score will never be higher than 0.2, even with an ideal retrieval system.

The precision score is easy to understand and compare but has to be critically evaluated if there are small minority classes or if the retrieval order is relevant. Furthermore, the multi-label similarity is strongly simplified and might lead to high values with qualitatively low performance. Especially for complex imagery such as remote sensing data, simply defining a single overlapping label as a relevant image might be too trivial.

Mean Average Precision

The mean Average Precision (mAP) is *not* identical to the overall precision score described in Eq. 5.5. mAP is defined as:

$$\text{mAP}@k(Q) = \frac{1}{|Q|} \sum_{q=1}^{|Q|} \text{AP}@k(q) . \quad (5.6)$$

Using the indicator (Eq. 5.4) and precision definition (Eq. 5.3), $\text{AP}@k(q)$ is given with

$$\text{AP}@k(q) = \frac{1}{\sum_{i=1}^k \mathbb{1}_{s(q,i)>0}} \sum_{i=1}^k \mathbb{1}_{s(q,i)>0} \times \text{precision}@i(q) . \quad (5.7)$$

For shorter notation, one can define $N_{\text{rel}} := \sum_{i=1}^k \mathbb{1}_{s(q,i)>0}$. Putting it all together, mAP can then be written as:

$$\text{mAP}@k(Q) = \frac{1}{|Q|} \sum_{q=1}^{|Q|} \frac{1}{N_{\text{rel}}} \sum_{i=1}^k \mathbb{1}_{s(q,i)>0} \times \text{precision}@i(q) . \quad (5.8)$$

Average precision calculates the mean precision scores from the first to the k -th retrieved images, while *masking* the non-relevant images. Due to the averaging procedure, mAP is less sensitive to the exact value of k compared to the classic precision score. Masking non-relevant images also allow the metric to become less sensitive to small minority classes.

Assume that there are only two images with a specific class l in the archive, but the archive contains thousands of other images. Setting $k = 10$ and using a query image with the class l , the average precision score can still become 1, in contrast to the classic precision score. For the average precision score to become 1, the non-relevant images have to be at the end of the retrieval queue. The metric

implicitly drops all non-relevant images at the end of the retrieval queue while giving higher weights to the retrieved images in the beginning.

Still, the metric has no *knowledge* of the remaining archive set. If a potential retrieval system only returns images with very high similarity confidence and then continues to return irrelevant images, the metric could become unexpectedly high. Tuning a system to the mAP metric could hide a bias where the system skips samples with lower confidence.

Also, for complex multi-label imagery defining a single matching label as an equally relevant image compared to a retrieved image that matches all labels might be not ideal. The relevance issue is inherited from the precision score.

Weighted Mean Average Precision

The weighted mean Average Precision (wmAP) score tries to encode similarity information among multi-label images better. The wmAP metric is based on the mAP and the average cumulative gain (ACG) metric [70]. The ACG score directly utilizes the similarity measure from Eq. 5.2 to allow for non-binary relevance values. These similarity values are then simply averaged to calculate the ACG score:

$$\text{ACG}@k(q) = \frac{1}{k} \sum_{i=1}^k s(q, i) . \quad (5.9)$$

To calculate the wmAP score the ACG from Eq. 5.9 replaces the precision metric in Eq. 5.8 which leads to the following definition:

$$\text{wmAP}@k(Q) = \frac{1}{|Q|} \sum_{q=1}^{|Q|} \frac{1}{N_{\text{rel}}} \sum_{i=1}^k \mathbb{1}_{s(q,i)>0} \times \text{ACG}@i(q) . \quad (5.10)$$

Like mAP, wmAP does not evaluate the remaining archive and cannot determine a possibly better retrieval order. In contrast to mAP, wmAP better integrates the multi-label information as it allows retrieved images to have varying degrees of *relevance*. Complex retrieval results give higher scores to query images with many labels.

Although the ACG allows for better encoding of relevant images, it introduces new issues. The overall score is not bound by 1 anymore and is biased towards many-label images. The maximum value is dependent on the underlying label distribution. The value itself has little meaning without any information about the underlying dataset. As a result, the wmAP score can only be compared against runs with identical query and archive splits.

The bias towards many-label images results from giving images with many labels a higher relevance than images with fewer. Correctly retrieving images for queries with 19 labels, for example, has a more significant influence on the

overall score than correctly retrieving images for queries with only three labels. To objectively evaluate a content-based image retrieval system, preferring specific image instances over others is an undesirable property.

Normalized Discounted Cumulative Gains

A different popular retrieval metric is the normalized discounted cumulative gains (NDCG) score [71]. It is based on the discounted cumulative gain (DCG) metric, which adds an order-based discount factor to ACG (Eq. 5.9). An order-based discount factor gives the retrieved images in the front of the queue a higher priority than those in the lower ranks. Mistakes further down the queue are not penalized as much, as the relevance of those images is lower. The commonly [72] used DCG definition is:

$$\text{DCG}@k(q) = \sum_{i=1}^k \frac{2^{s(q,i)} - 1}{\log_2(1 + i)}. \quad (5.11)$$

To make the DCG scores comparable to other query results and not dependent on the data distribution, a normalization step is added to define the NDCG metric for a single query q :

$$\text{NDCG}@k(q) = \frac{\text{DCG}@k(q)}{\min(\text{IDCG}@k(q), 1)}, \quad (5.12)$$

where IDCG is the ideal discounted cumulative gain of the complete archive set that encodes the highest possible discounted cumulative gain score. Note that the IDCG score needs to have a lower bound of 1 to ensure that NDCG is well-defined even if there are no matching images in the archive set.

Due to the normalization, the NDCG metric is cross-query comparable and can be extended to the entire query set:

$$\text{NDCG}@k(Q) = \frac{1}{|Q|} \sum_{q=1}^{|Q|} \frac{\text{DCG}@k(q)}{\min(\text{IDCG}@k(q), 1)}. \quad (5.13)$$

The NDCG metric is bound by the number 1 and can be easily compared with different query/archive splits. The score takes the retrieval order into account and has full *knowledge* of the archive, as the highest possible DCG value is used as a normalization factor. As a result, the value of k is not bound by any class, nor is the value biased towards many-label images.

One could argue that the NDCG metric requires too many compute resources for vast archives. However, with the ever-increasing capabilities of modern hardware and software that allow for out-of-memory calculations [73], NDCG scores for archives with millions of entries can be quickly calculated.

An arguable disadvantage of NDCG, which is also shared with all of the presented metrics, is the use of the similarity definition from Eq. 5.2. The provided similarity definition only considers the intersection between the label sets of the query and archive image. There is no *penalty* if one label set is considerably larger than the other. An appealing alternative would be to define a new similarity function, where the label sets are ordered multi-hot encoded vectors, and \odot defines the logical XNOR operation:

$$s(q, i) = \sum (L_q^* \odot L_i^*) \quad . \quad (5.14)$$

Although a thorough analysis of the proposed similarity measure is not in the scope of the thesis, it should be noted that it is crucial to keep the implications and biases of the used metrics in mind when evaluating benchmark results.

Compared to the other image retrieval metrics, the NDCG score has the most advantages for complex multi-label imagery such as satellite data. The NDCG metric does not simplify the multi-label information, as precision or mAP do. NDCG is not biased towards many-label images, in contrast to wMAP. It evaluates the performance with respect to the best possible results from the archive and produces easy to evaluate and compare values between 0 and 1.

Therefore, the NDCG will be regarded as the *primary* metric to evaluate the performance of the following experiments. The other metrics will also be included in the evaluation. However, the NDCG will serve as the core indicator of content-based image retrieval performance for remote sensing data.

5.3 Default Experimental Setup

To minimize the differences between the following experiments and the recommendations from the previously proposed CSSRL methods, the main neural network architecture will be the residual network (ResNet) [21].

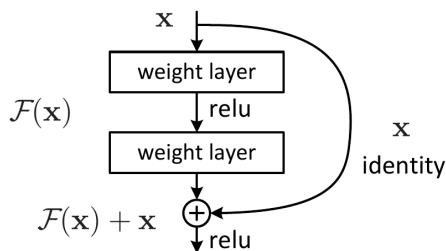


Figure 5.6: Skip Connection Visualization (Image from [21])

He et al. [21] proposed the use of *skip connections*, shown in Fig. 5.6. These skip connections allow building deeper models without decreasing the performance while keeping the number of parameters and computation complexity the same.

Since the initial proposal of the ResNet architecture, research has improved the initial architecture by adding new [74, 75] or tweaking [76] existing components.

However, to compare results with a wide range of previously published papers, most CSSRL methods evaluate their model on the original ResNet architecture. For the same reason, the chosen optimizer is the Adam optimizer [77] with cosine annealing after 75 % of the total epochs. The default configuration for the following experiments is: $\text{lr} = 1 \times 10^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.99$, and $\varepsilon = 1 \times 10^{-5}$. These values were chosen after running a hyperparameter grid search on smaller subsets.

The default augmentation pipeline is motivated in Chapter 3 and consists of:

- Dihedral Transformation
- Rotation
- Gaussian Blurring
- Resized Cropping

The specific values for the various augmentation techniques are identical to the proposed values from Chen et al. [41]. Concretely this means that the blur kernel size is 10 % of the original image width/height rounded up to an uneven number. The resulting kernel size is 13 px×13 px for the BigEarthNet dataset. The blur factor is randomly sampled for each image from the uniform distribution $\sigma \in [0.1, 2.0]$. The resized crop augmentation crops a randomly sized part (from 8 % to 100 % in area from original image) with a random aspect ratio (between 3/4 and 4/3). The rotation transformation randomly rotates the input between 0° and 45°. Each augmentation technique, except for random cropping, is applied to each branch with a probability of 50 %. Note that the symmetric augmentation probabilities are used for SimCLR, Barlow Twins, and BYOL.

Although not part of the recommended augmentation pipeline, the multi-spectral color-jittering augmentation (composition of brightness and contrast transformation; see Chapter 3) defines a unified hyperparameter `max_lighting`. The experimental results chapter will evaluate the effect of the simplified color-jittering augmentation and requires the unified hyperparameter `max_lighting`. The `max_lighting` parameter is bound between [0, 1] and defines the relative strength of the brightness and contrast transformation. A value of 0 is equal to no transformation applied at all and a value of 1 to the maximum brightness/contrast transformation. From the available twelve multi-spectral channels, all but the two low-resolution 60 m channels were used.

The default batch size is set to 512, which is low compared to the recommendations from the proposed CSSRL methods. The small batch size was chosen to easily allow testing of larger and more complex architectures and, more importantly, for better comparisons between the *standard* and guided sampling experiments. The experiments were run for 41 epochs to keep the overall training time low, even if the general CSSRL recommendation is to train for hundreds of epochs [41, 43, 44]. The experimental results chapter evaluates the impact of the comparatively small batch size and training time.

The metadata-guided sampling strategy experiments investigate the effect of 10, 16, 64, 128, 256, and 512 clusters. The general statistics of these cluster configuration were presented in Chapter 4 Table 4.1. The 128, 256, and 512 cluster

configurations were mainly chosen as recommended values for the mixed-cluster sampling strategy. As the mixed-cluster sampling strategy derives the batch size from the number of clusters, the 512 cluster configuration allows to directly compare the results of the default and mixed-cluster sampling strategy without being constrained by the cluster sizes.

Fig. 5.7 visualizes the cluster results for 128 and 512 clusters. The visualization shown in Fig. 5.7b could suggest that 512 clusters may be too fine-grained. The smaller cluster configurations (128 and 256) allow the cluster sizes to increase and group the data into *softer*, more general clusters.

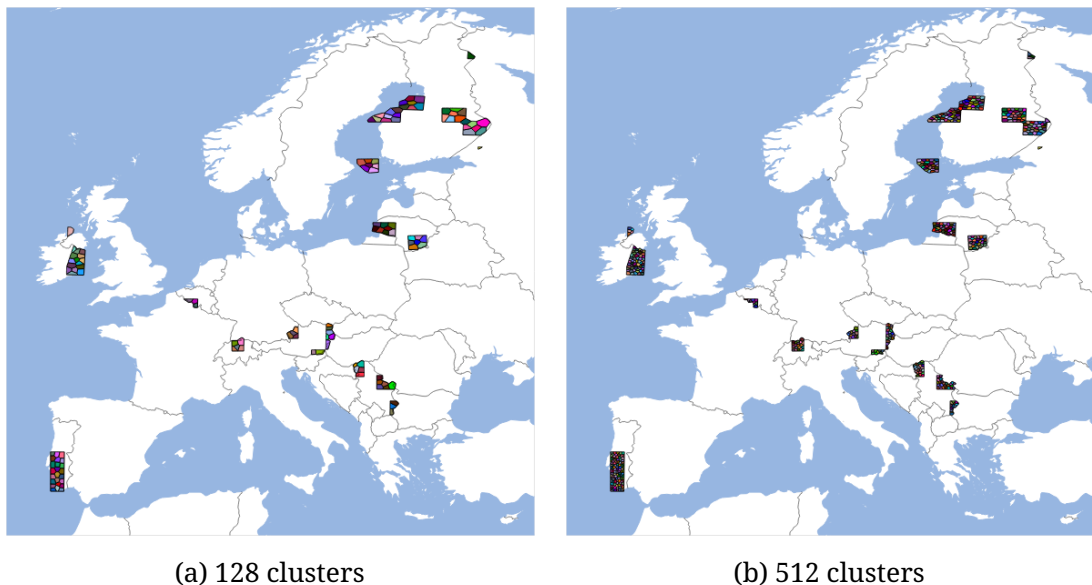


Figure 5.7: BigEarthNet-Summer cluster visualizations with 128 and 512 clusters.

These cluster configurations can also be used to create hard batches with the in-cluster sampling strategy. The main issue is that the batch size is constrained to 72 or 57 patches, which might hurt the performance of the CSSRL methods. The cluster configurations with 10, 16, and 64 clusters group the data into larger clusters without too heavily restricting the batch size. The visual results are presented in Fig. 5.8.

The code for the proposed metadata-guided sampling strategy, as well as the code for all of the following experiments, is available at <https://git.tu-berlin.de/rsim/self-supervised-cbir-with-smart-sampling>. The code is build on-top of the `fastai` library from Howard and Gugger [78] and the `self_supervised` library from Turgutlu et al. [79]. Part of the thesis' code has been merged upstream for broader impact of the current work (see <https://github.com/fastai/fastai/pull/3252>, <https://github.com/fastai/fastai/pull/3255>, and https://github.com/KeremTurgutlu/self_supervised/pull/19).

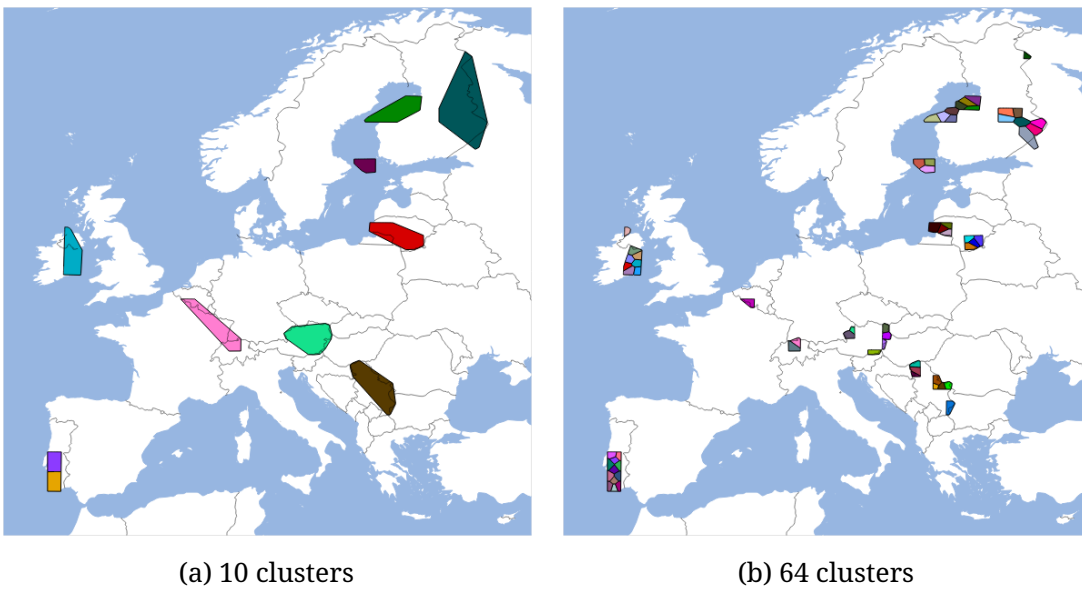


Figure 5.8: BigEarthNet-Summer cluster visualizations with 10 and 64 clusters.

6 RS CBIR Results

Every experiment evaluates the NDCG, mAP, wmAP, and the precision performance for the first $k = 5, k = 10, k = 20, \dots, k = 100$ retrieved images. These specific values were copied from Sümbül and Demir [80]. Note that the chosen BigEarthNet-Summer archive is big enough not to constrain the precision or mAP score. For both of these metrics, the highest possible value remains 1.0/100 %.

Aside from using metrics, it would be possible to evaluate the qualitative performance. The main issue is that it is only feasible to review a small fraction of the retrieved samples due to the query and archive split size. Furthermore, due to the high complexity of satellite imagery, it is not easy to evaluate the performance by visual inspection. To better support the argument, two retrieval results are shown in Fig. 6.1 without stating which model is a randomly initialized and which a self-supervised trained model¹. Due to the issues with the qualitative evaluation method, the following chapter will only focus on evaluating the quantitative performance of the presented CSSRL methods.

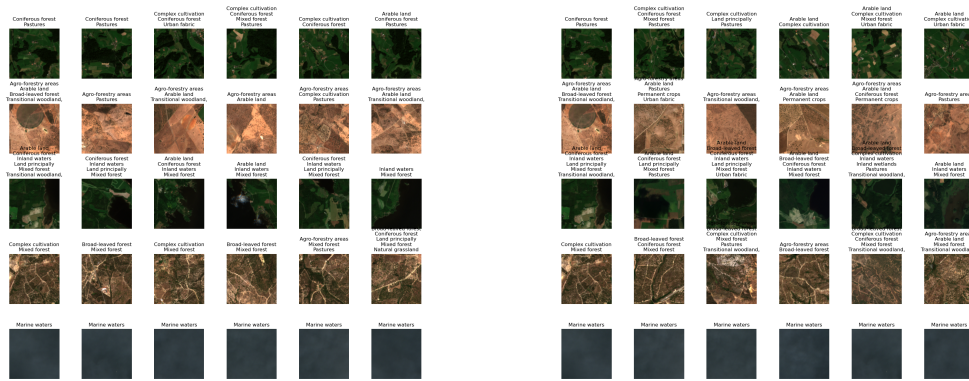
6.1 Analysis of Investigated CSSRL Methods

All presented CSSRL methods — SimCLR, Barlow Twins, and BYOL — are tested with their recommended projection head and the proposed augmentation pipeline. A supervised and a randomly initialized model were benchmarked in addition to the presented CSSRL methods for a more general comparison. The score for each IR metric can be seen in Fig. 6.2.

As expected, the supervised model performs the best, and the randomly initialized model performs the worst. Since the supervised model can directly learn from the ground-truth labels, it is able to learn finer features. The relatively high performance of the randomly initialized model is due to the convolutional structure, which gives a strong prior on the input signal [37]. Note, these results highlight that the precision and mAP scores are generally high (close to 1.0) for complex multi-label datasets and that even a random model can get scores over 90 %.

There seems to be no CSSRL method that is out-performing the others. Looking at the NDCG results, SimCLR performs best for few retrieved images, while BYOL takes the lead if more than twenty images are retrieved.

¹ Fig. 6.1a is the SimCLR trained model and Fig. 6.1b is the randomly initialized model



(a) Retrieval System A

(b) Retrieval System B

Figure 6.1: Example retrieval results of a randomly initialized and a self-supervised trained model. The first column shows the query images, followed by the five most similar retrieval results. Which retrieval system corresponds to the randomly initialized and which to the self-supervised one is not shown to highlight the qualitative similarity between both systems.

It is important to note that the labels of the retrieved images have a large effect on the NDCG score, especially if only a few images are retrieved. As a result, the NDCG score for fewer images may fluctuate from run to run. With this in mind, BYOL seems to be the favorable CSSRL method, which is in line with the results from the natural image domain [43].

Since most experiments show a high correlation between the precision, mAP, and wmAP scores, only the precision and NDCG scores will be presented. Limiting the results to two metrics per experiment reduces the visual noise. The complete benchmark results are available at <https://git.tu-berlin.de/rsim/self-supervised-cbir-with-smart-sampling>. Experimental results that mainly support conclusions or add little value from previous experiments have been added to Appendix B.

The previous results used the recommended augmentation pipeline for remote-sensing images proposed in Chapter 3. Since augmentation strategies play a crucial role in CSSRL methods, the proposed pipeline needs to be critically analyzed.

The general issue with hyperparameter-based analysis is that the experimental setup quickly suffers from a combinatorial explosion. The strategy from Chen et al. [41] was applied to limit the number of experiments to a feasible amount. Every experiment only changes a single hyperparameter or disables/enables a specific augmentation strategy of the proposed augmentation pipeline.

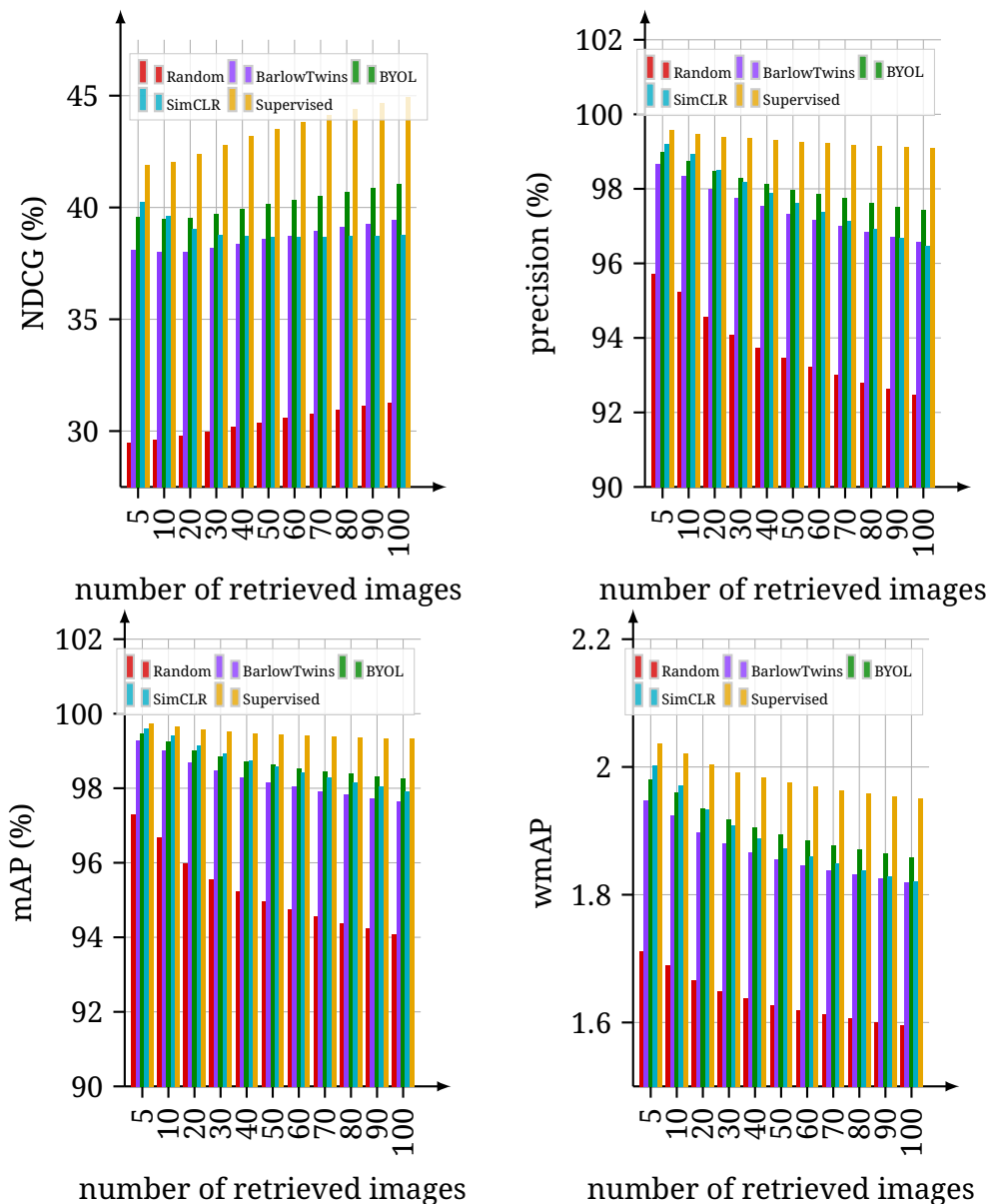


Figure 6.2: Default benchmark results with the presented CSSRL methods (Barlow Twins, BYOL, SimCLR), a randomly initialized model, as well as a model trained with supervision.

Simplified Multi-Spectral Color-Jittering

The CSSRL methods may suffer from the missing color-jittering augmentation. Experiments from the natural image domain have shown that SimCLR may apply histogram matching to find the correct positive pairs if no color-jittering is applied

[41], leading to subpar performance. As described in Chapter 3, the closest multi-spectral composition to SimCLR’s color-jittering is a combination of brightness and contrast augmentation. The effect of varying simplified color-jittering strengths on the SimCLR method can be seen in Fig. 6.3 (all metrics are shown in Fig. B.1).

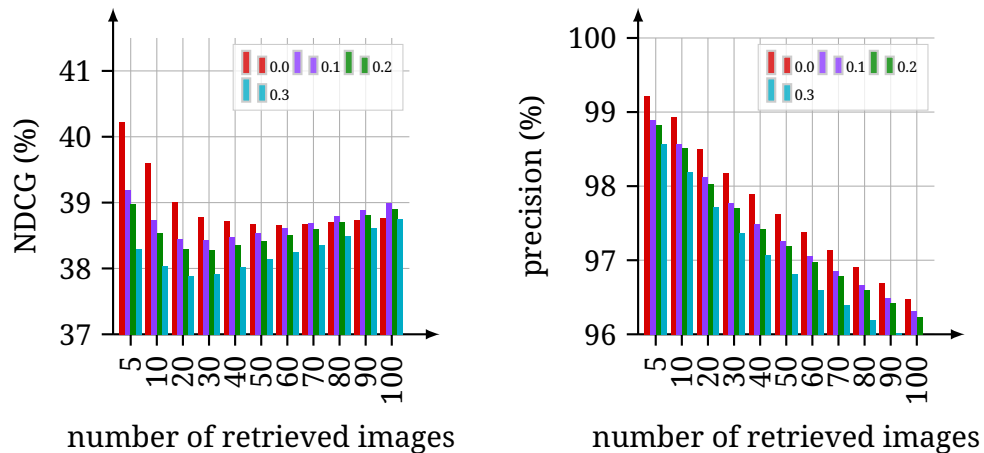


Figure 6.3: Simplified multi-spectral color-jittering augmentation (brightness + contrast shifting) results with SimCLR method and different `max_lighting` values.

The precision score is best when no color transformation is applied and gets worse with stronger color augmentations. The NDCG score indicates that the first thirty retrieved images are much more *accurate* when no color augmentation is applied but becomes worse when more images are retrieved. However, the NDCG score strongly fluctuates between runs. Either the NDCG score is comparatively high for smaller amounts of retrieved images and gets worse for larger amounts, as seen in Fig. 6.3, or the performance benefit for fewer retrieved images is smaller but higher for all different retrieval settings when no color augmentation is applied. The fluctuation only happens for the NDCG score; the precision, mAP, and wmAP scores worsen for stronger color augmentations. These fluctuations are only present in the SimCLR experiments and are reviewed in more detail later.

The Barlow Twins and BYOL methods paint a more concrete picture. The Barlow Twins results are shown in Fig. 6.4 (complete results in Fig. B.2) and the BYOL results in Fig. 6.5 (complete results in Fig. B.3). For both of these methods, the color augmentation transformation considerably hurts the overall retrieval performance. In general, these results indicate that the initial recommendation of not using any color augmentation techniques for CSSRL methods in the remote sensing domain is correct.

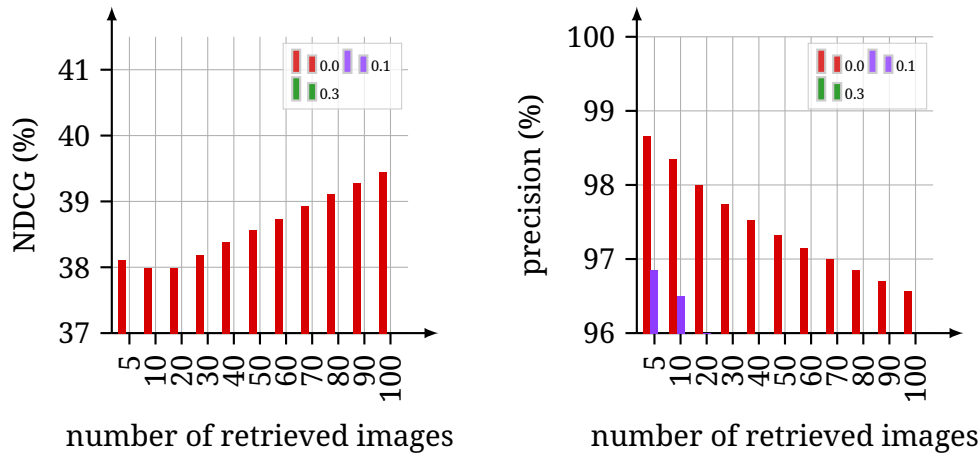


Figure 6.4: Simplified multi-spectral color-jittering augmentation (brightness + contrast shifting) results with Barlow Twins method and different `max_lighting` values.

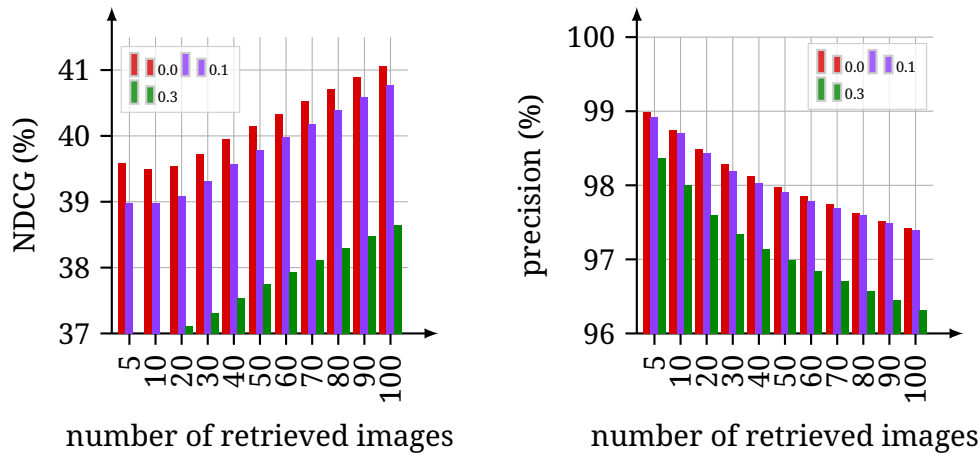


Figure 6.5: Simplified multi-spectral color-jittering augmentation (brightness + contrast shifting) results with BYOL method and different `max_lighting` values.

Dihedral Transformation

The proposed augmentation pipeline strongly recommends applying a Dihedral transformation since remote sensing images are rotation invariant. The experimental results do not fully align with the initial assumption. BYOL and SimCLR slightly favor a Dihedral transformation (see results in Figs. B.4 and B.5), while the Barlow Twins method takes a significant performance hit, as shown in Fig. 6.6.

The relatively weak benefit for BYOL and SimCLR suggests that either the random 45° rotation suffices, or that the rotation invariance is not too relevant to

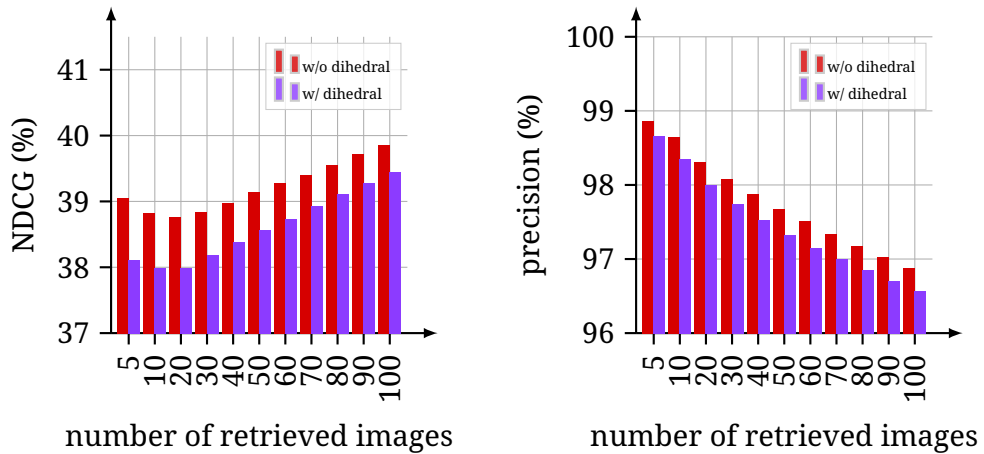


Figure 6.6: Dihedral transformation results with Barlow Twins method.

generate *good* contrastive pairs. The strong negative impact of flipping and rotating on the Barlow Twins’ retrieval performance is unexpected and requires further investigation. A pointer for further investigation is sketched in Fig. 6.7, which shows that the Barlow Twins method performs best when no Dihedral transformation is applied but performs worse when neither the rotation nor the Dihedral transformation is used.

In general, the Dihedral transformation should be included in the CSSRL pipeline for RS imagery. Since rotational invariance is a fundamental property of remote sensing data, the model should almost always *learn* this property. The major exception is the Barlow Twins method, which seems to suffer from the Dihedral transformation. A possible explanation for the low benefit for SimCLR and BYOL could be that the rotation invariance is not too relevant due to the high correlation between query and archive split. The effect of randomly rotating the query or archive images could be investigated to test this assumption further.

Gaussian Blurring

Gaussian blurring behaves similarly to the Dihedral transformation. SimCLR and BYOL both prefer Gaussian blurring, as shown in Figs. B.6 and B.7, while the Barlow Twins (Fig. 6.8) favor no blurring at all. Although Zbontar et al. [44] have used the same Gaussian blurring configuration in the original publication. Gaussian blurring seems to give minor improvements for SimCLR and BYOL but requires some tuning for the Barlow Twins. Either by completely disabling blurring or by investigating if smaller blur strengths or kernel sizes help to improve the performance. However, blurring does not force the model to learn any desirable properties, such as the Dihedral transformation enforces rotation invariance. Blurring increases the input variability and, potentially, makes the resulting model less reliant on edges. If higher performance is achievable without blurring, it is

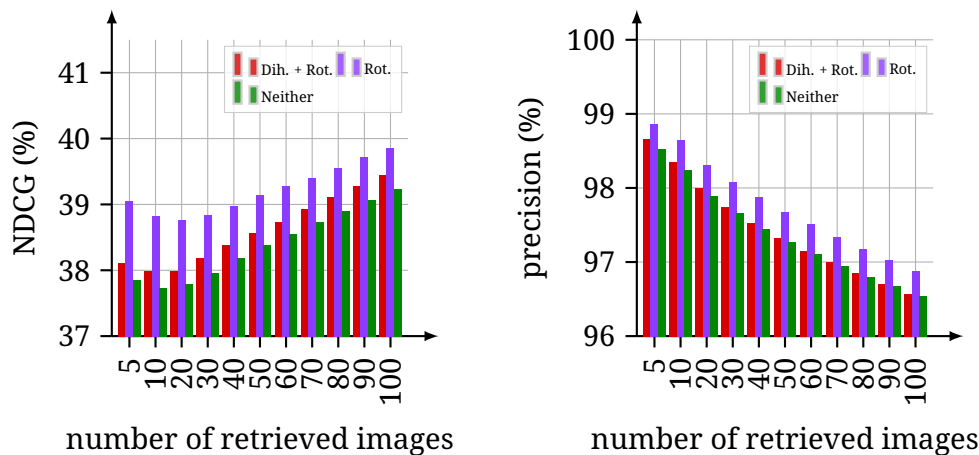


Figure 6.7: Effect of rotation and Dihedral transformation on the Barlow Twins method.

an equally valid consideration.

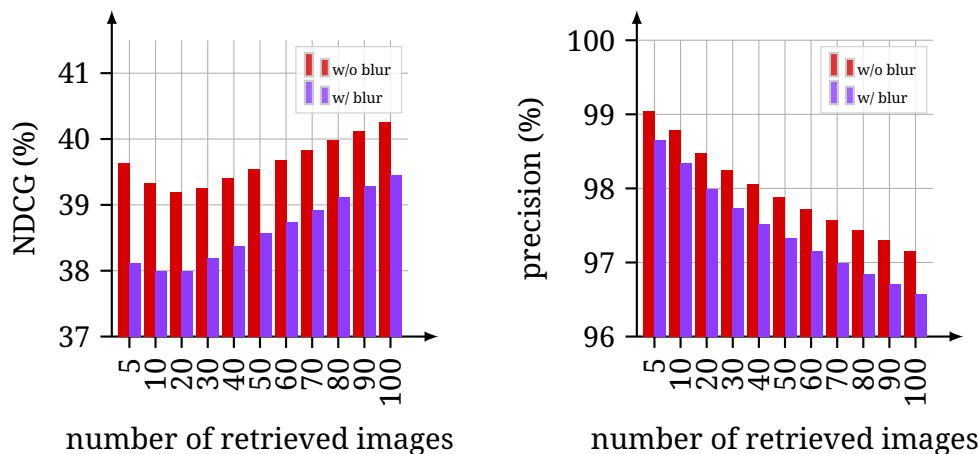


Figure 6.8: Gaussian Blurring results with Barlow Twins method.

Resized Cropping

A possibly questionable design decision done in the default augmentation pipeline was to set the minimum size of the crop before resizing to 8%. Due to the high-spatial resolution of each pixel, smaller crops are likely to drop entire classes from input images. The desired effect on the model is to allow it to become scale-invariant and to introduce more variability to the input. The higher variability may also counter possible *short-circuit* solutions, like histogram matching. Another reason to consider allowing smaller crops is that the results from the natural

image domain strongly suggest doing so.

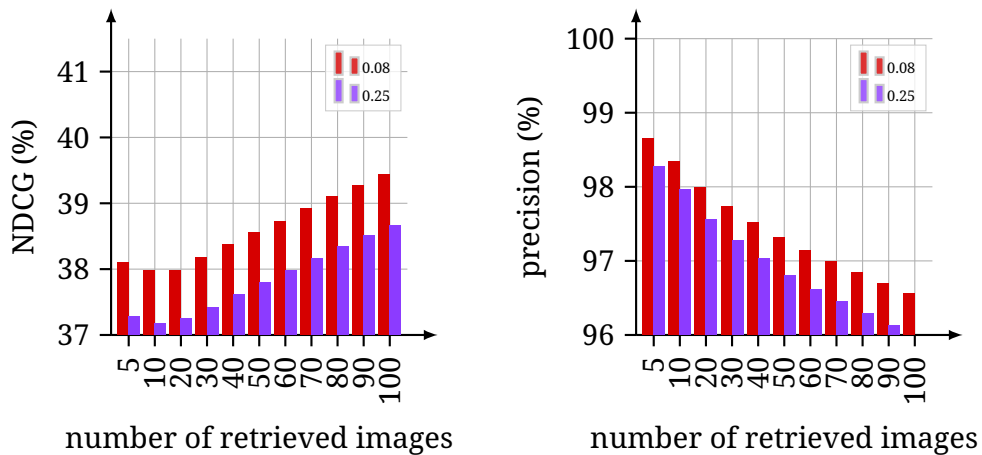


Figure 6.9: Resized cropping results with Barlow Twins method and different minimum crop sizes.

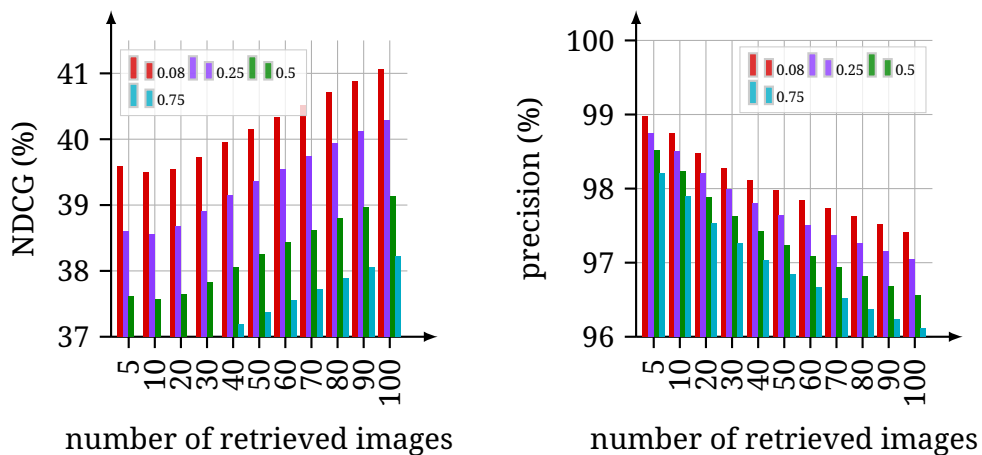


Figure 6.10: Resized cropping results with BYOL method and different minimum crop sizes.

The experimental results shown in Figs. 6.9 and 6.10 support the choice of small crop sizes. The Barlow Twins and the BYOL methods perform considerably worse with larger crop sizes. The SimCLR results (see Fig. B.8) also generally favor smaller crop sizes. However, similar to the color-jittering experiments, SimCLR produces inconsistent results and, generally, prefers larger crops if more than thirty images are retrieved. Still, the overall recommendation is to use smaller crop sizes for CSSRL methods, as suggested in the initial augmentation pipeline proposal.

Investigating SimCLR's Instability

As previously mentioned, the SimCLR results fluctuate relatively strong compared to the Barlow Twins or BYOL method. To gain more insight into the training dynamics, the training was stopped every ten epochs to calculate the retrieval scores over time. Comparing the NDCG scores over time from all three CSSRL methods shows that the issue is not caused by training for a relatively short amount of time. The Barlow Twins and BYOL method generally get higher NDCG scores the longer the training runs with diminishing returns.

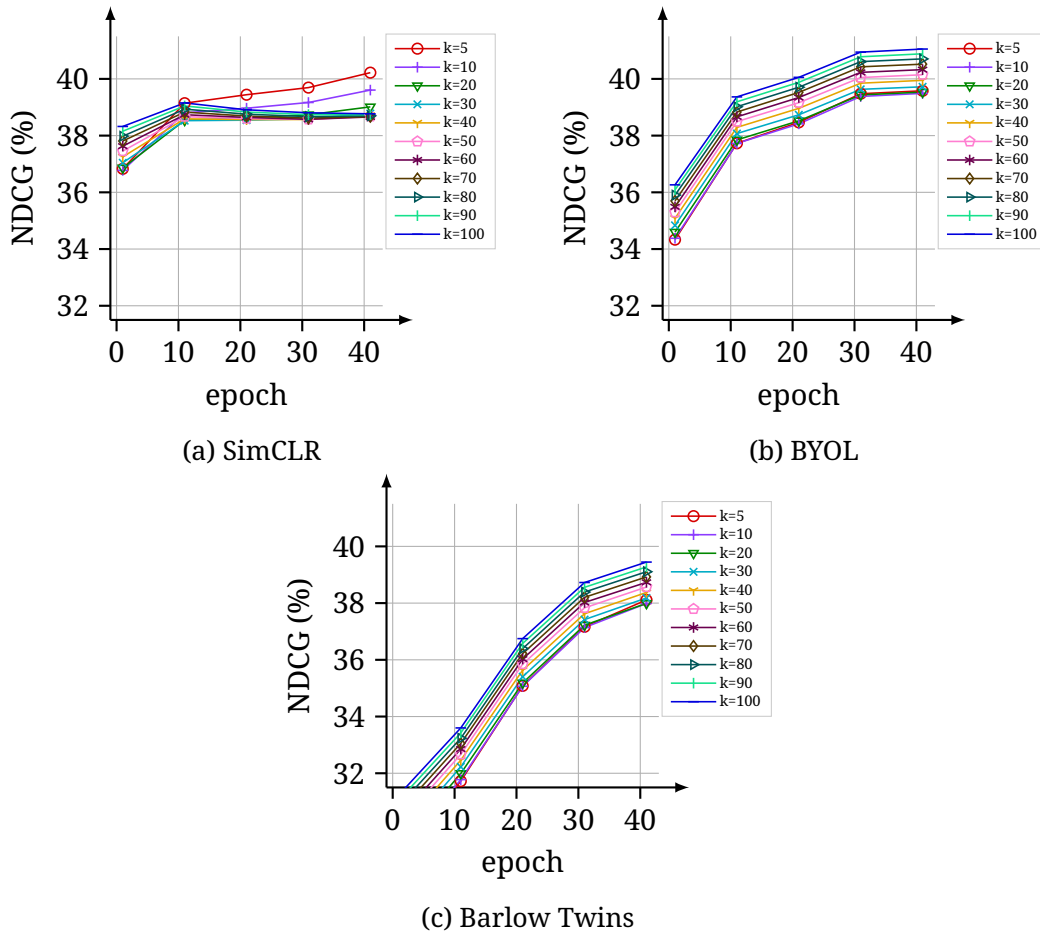


Figure 6.11: Intermediate NDCG scores for SimCLR, BYOL, and Barlow Twins during training.

SimCLR, on the other hand, starts to perform worse after twenty epochs of training if more than ten images are retrieved. Conversely, the scores for five to ten retrieved images strongly increase. The high scores for few retrieved images indicate that the model is overfitting specific views. Due to the strong correlation between the query and archive split of the BigEarthNet dataset, there is a high

likelihood of *very* similar images in both splits. SimCLR seems to learn patch-specific features that do not generalize well. These features allow the model to find strongly correlated images, leading to a good performance if only a few images are retrieved. When more images are retrieved, the effect of non-generalizable features becomes apparent and reduces the overall result.

This hypothesis would explain the fluctuations seen in the SimCLR experiments. Some runs start to overfit early, while others seem to overfit later. Over fitted methods lead to strong performance for few retrieved images, but low performance when more images are retrieved, and why strong augmentation techniques help SimCLR perform better when more images are retrieved. The performance does not necessarily improve due to the *better* fit of the augmentation pipeline in general, but because it offsets the over-fitting effect.

SimCLR could suffer from the relatively small batch size of 512. However, the results shown in Fig. 6.12 do not fully support this argument. Larger batch sizes help stabilize the results in some experiments and reduce the effect of overfitting but do not solve the underlying issue. At the end of the training with 2048 patches per batch, as shown in Fig. 6.12b, the model starts to overfit, and the scores for $k > 20$ start to decrease and collapse. Furthermore, the overall scores are generally *not* higher for larger batch sizes. A non-increasing performance for larger batch sizes is in contrast to results from the natural image domain.

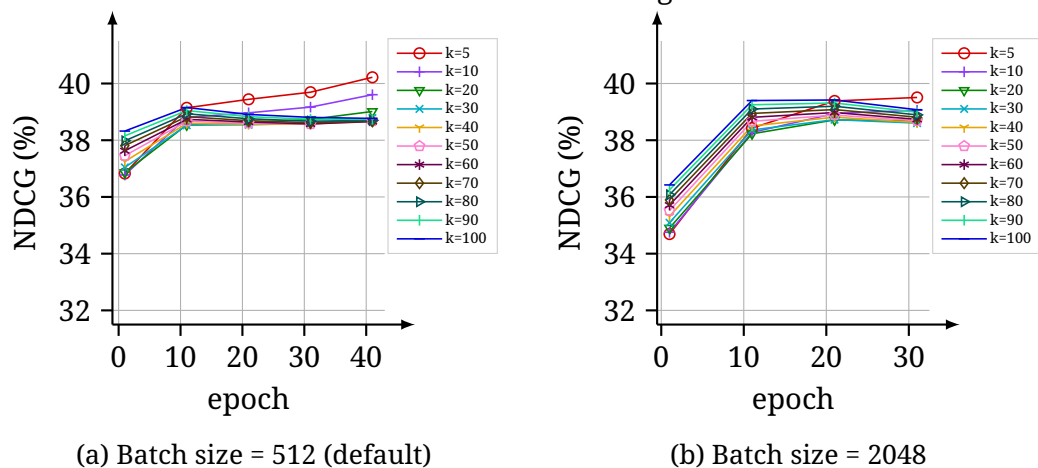


Figure 6.12: Intermediate NDCG scores for SimCLR with different batch sizes.

Since the SimCLR results are unstable and do not seem to overfit in all experiments, another possible solution is to use a different optimizer. A better optimizer would be a variant that helps to avoid early local minima and encourages exploration. To test the assumption, the Adam optimizer was replaced with a novel synergistic optimizer: *Ranger21* [81]. The NDCG scores over time can be seen in Fig. 6.13.

The results support the assumption that the *Ranger21* optimizer helps to avoid the overfitting issue. The NDCG scores for $k > 10$ with the *Ranger21* optimizer

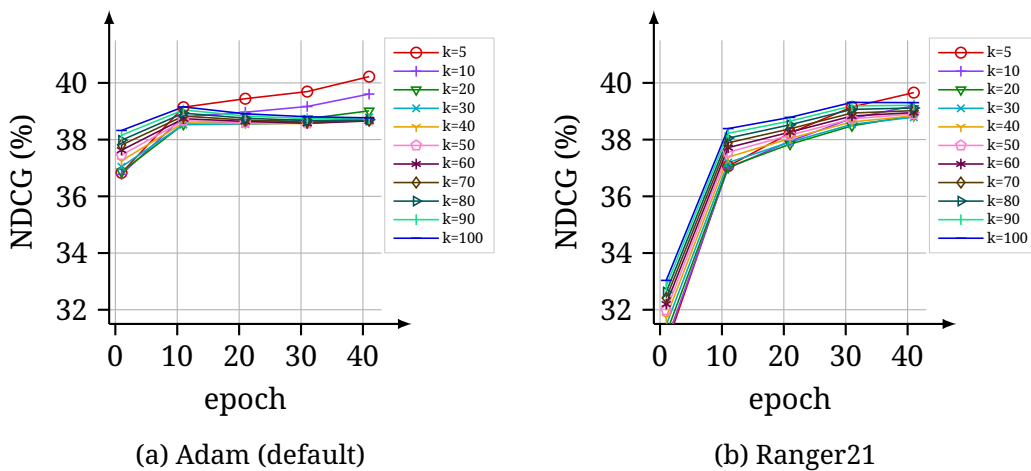


Figure 6.13: Intermediate NDCG scores for SimCLR with default Adam and Ranger21 optimizer.

converge more smoothly and do not worsen over time. Note that the scores for $k < 10$ are lower than the Adam optimizer scores. However, since the model from the Adam optimizer is overfitted, it is to be expected that a more general model performs worse for few retrieved images. Especially if the query and archive split are heavily correlated, as it is the case for the current BigEarthNet split. In general, the SimCLR results are much more stable with the Ranger21 optimizer. With the new optimizer, the previous contradictory results for SimCLR are also *fixed*. The SimCLR color-jittering and resized-crop results are now in line with BYOL and the Barlow Twins results. As shown in Fig. 6.14, adding any type of color augmentation greatly hurts the overall performance, as well as large crop sizes as shown in Fig. 6.15.

The class imbalance in the dataset combined with the way SimCLR contrasts the samples might explain the tendency of SimCLR to overfit. The current training split of the BigEarthNet-Summer dataset contains 65 599 patches with 3766 unique labels combination, which might indicate a high variability. However, of these 3766 label combinations 1627 exist once and 9970 ($\approx 15\%$) belong to the largest class combination, which only consists of the “Marine Waters” class.

This specific class imbalance could be devastating for the SimCLR training process. If a batch contains many semantically similar samples, such as the “Marine Waters” class, SimCLR might not be able to find the correct positive pairs due to the high correlation between water patches, or it learns water-specific features that do not generalize well. Both of these scenarios could lead to the observed, unstable behavior of SimCLR.

BYOL only contrasts positive pairs against each other and will probably be biased towards the largest class, but it should not trivialize the learning procedure. The Barlow Twins method does not contrast samples directly against each other but optimizes the cross-correlation between the metric embedding matrices. Both of

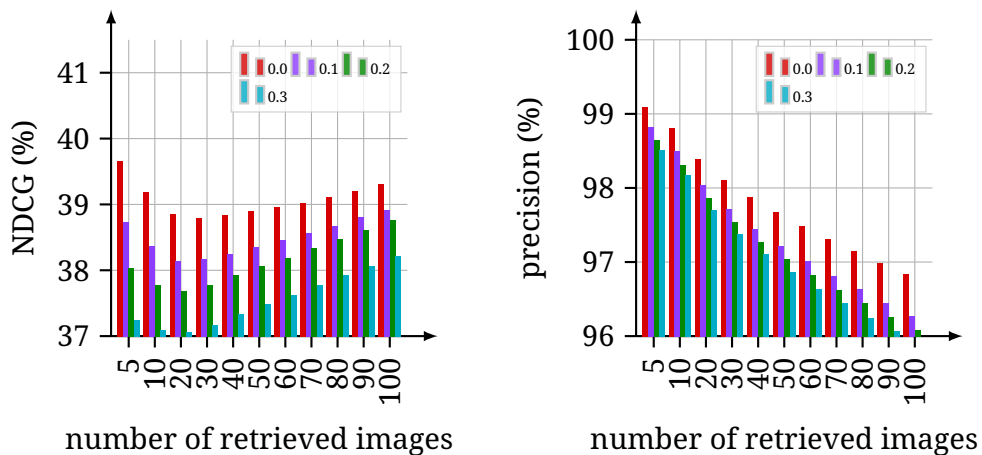


Figure 6.14: Simplified multi-spectral color-jittering augmentation (brightness + contrast shifting) results with SimCLR method + Ranger21 optimizer and different `max_lighting` values.

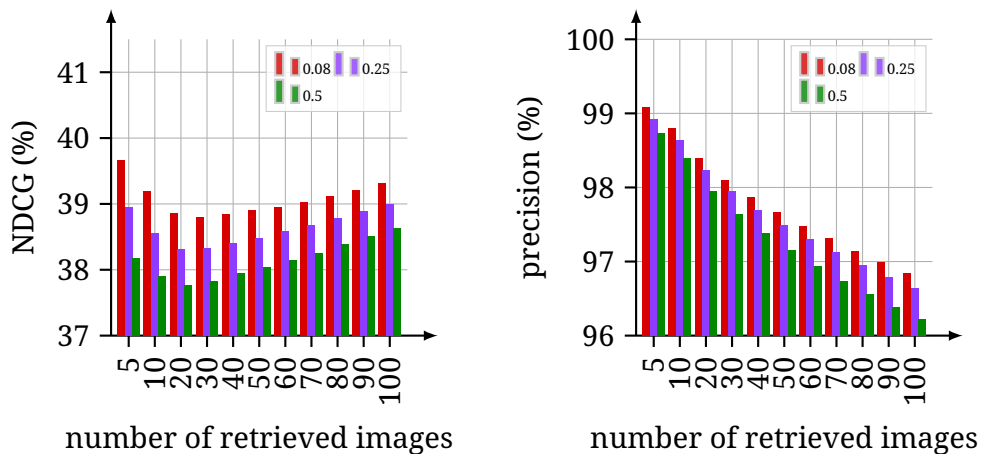


Figure 6.15: Resized cropping results with SimCLR method + Ranger21 optimizer and different `max_lighting` values.

these optimization methods seem to stabilize training even with imbalanced data.

Effect of Ranger21 on Barlow Twins and BYOL

The Ranger21 optimizer also had beneficial effects on the Barlow Twins method. The baseline results have improved considerably, as shown in Fig. B.9. This might be due to the *new* stability against the Dihedral transformation (see Fig. B.10). In contrast to the results with the Adam optimizer, the Ranger21 results perform almost identical with/without Dihedral transformation. However, the Barlow Twins still prefer no blurring at all, as depicted in Fig. B.11. The overall perfor-

mance of BYOL decreases when the Ranger21 optimizer is used (Fig. B.12). The performance gap can be reduced when the Ranger21 optimizer is run for more epochs to compensate for the slower convergence speed of the Ranger21 optimizer compared to the Adam optimizer.

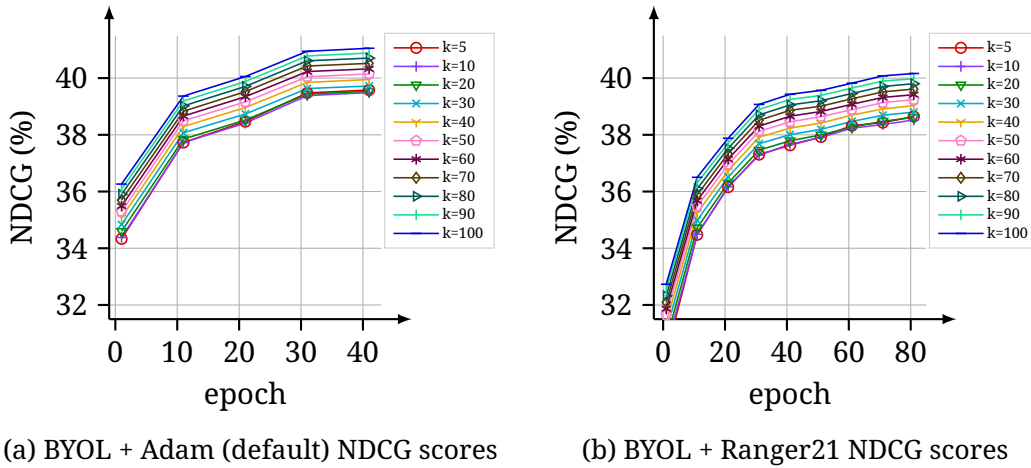


Figure 6.16: Intermediate NDCG scores for BYOL with default Adam and Ranger21 optimizer.

The results shown in Fig. 6.16 indicate that the BYOL and Ranger21 combination benefits from longer training runs but is generally lower than the Adam counterpart. Longer training runs or further tuning the optimizer's hyperparameter values might close the Ranger21 and Adam optimizer gap. In general, the benefits of the Ranger21 optimizer outweigh the drawbacks. Especially the stability improvements related to SimCLR are crucial for further analysis. Hence, the Ranger21 optimizer is used for *all* the following experiments.

Since the performance of SimCLR heavily depends on how the patches within a batch relate to one another, SimCLR has a high potential to benefit from the previously proposed metadata-based sampling framework from Chapter 4. The following section will investigate the effects of the proposed sampling framework on the different CSSRL methods. The experimental results from this section have shown that the initially proposed contrastive data augmentation pipeline for remote sensing imagery works very well with the presented CSSRL method. The recommended pipeline will be used for all of the remaining experiments. Even though the Barlow Twins prefer no Gaussian blurring, using the same pipeline for all methods minimizes possible side effects.

6.2 Analysis of Proposed Metadata-Guided Sampling Framework for CSSRL

The main motivation for the proposed metadata-based sampling framework from Chapter 4 is to tune to the batch *hardness*. According to Tobler’s first law of geography, the proposed in-cluster sampling strategy generally increases the batch hardness, while the mixed-cluster sampling strategy reduces it.

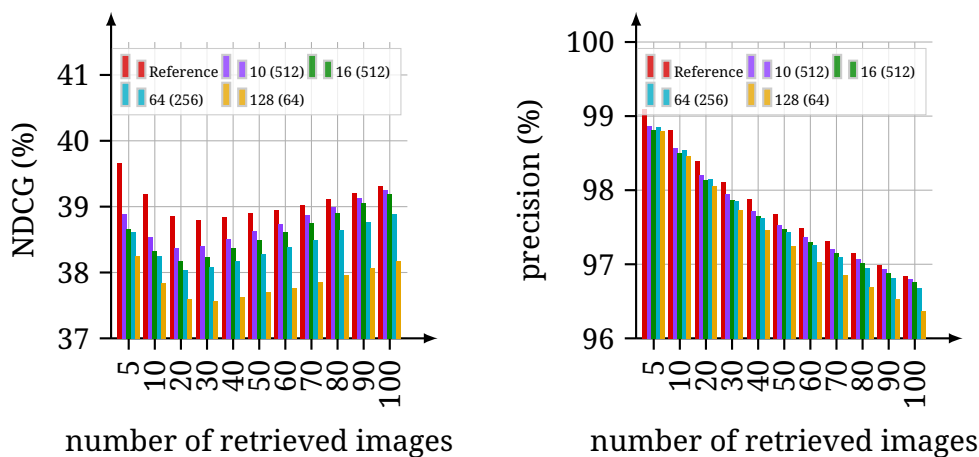


Figure 6.17: SimCLR in-cluster sampling results with c clusters and the batch size in parenthesis.

Increasing the batch hardness has an increasingly negative effect on SimCLR, as shown in Fig. 6.17. The significant performance drop with ten clusters (and a batch size of 512) indicates that the sampled batches are *too hard*. A contra-argument could be that the model requires more time to learn finer features due to the higher similarity among the patches within a batch. The progression results shown in Fig. B.13 disprove the assumption.

Decreasing the batch hardness by applying the mixed-cluster sampling strategy may improve the overall performance of SimCLR, as shown in Fig. 6.18. The retrieval performance is consistently higher with 512 clusters and therefore 512 patches per batch. The models trained with fewer clusters overfit to limited views and generally perform worse than the reference result. These results support the previous conclusion that batch size plays an essential role in the stability of SimCLR. The sampling strategy alone is not able to alleviate the overfitting issue. However, the proposed metadata-guided sampling strategy can improve the overall retrieval performance.

The results of the Barlow Twins method paint a similar picture. The in-cluster sampling strategy consistently hurts the overall retrieval performance, as shown in Fig. 6.19. Increasing the number of clusters worsens the penalizing effect of higher similarity within batches.

6.2 Analysis of Proposed Metadata-Guided Sampling Framework for CSSRL

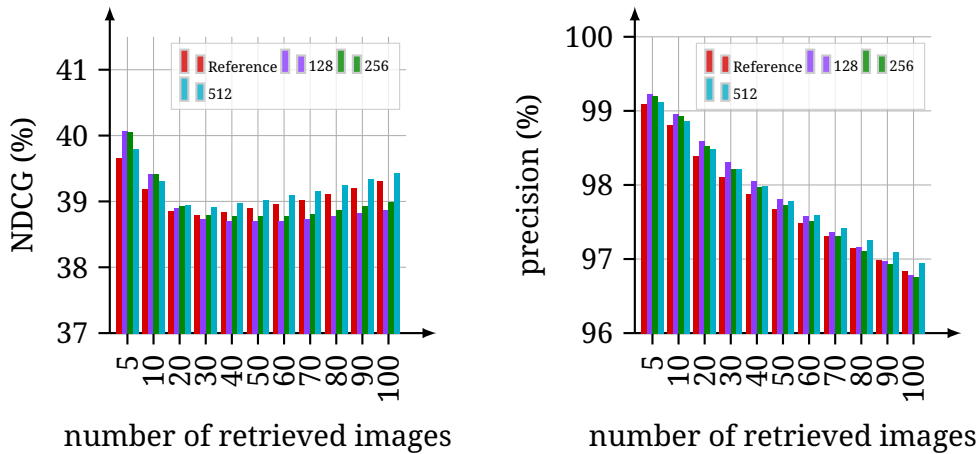


Figure 6.18: SimCLR mixed-cluster sampling results with c clusters.

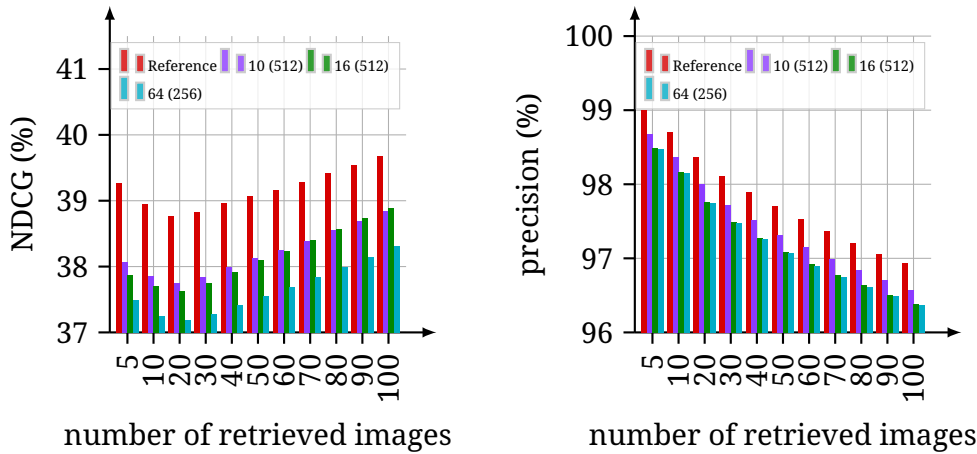
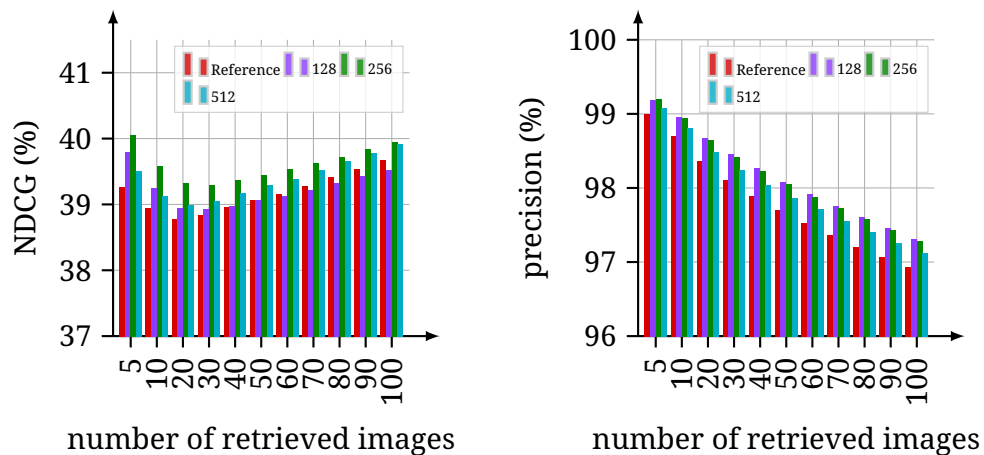
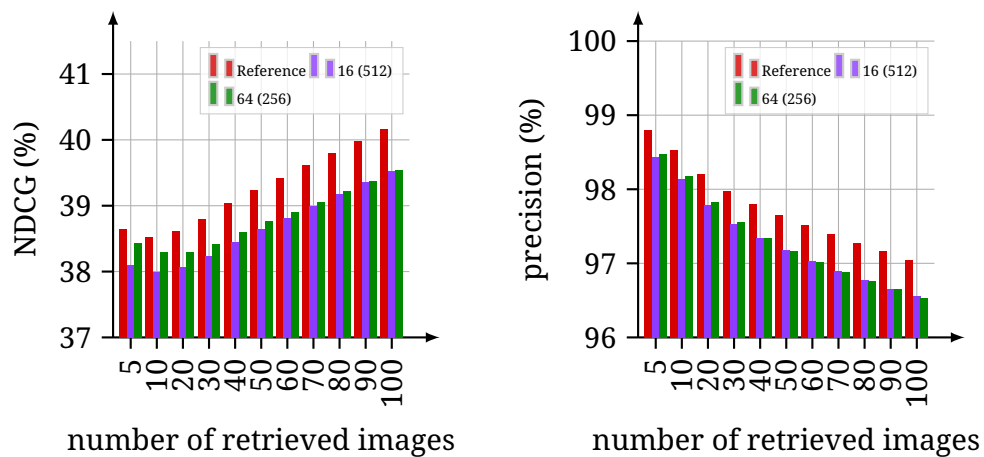


Figure 6.19: Barlow Twins in-cluster sampling results with c clusters and the batch size in parenthesis.

By mixed-sampling the clusters and creating batches with a higher variability, the performance generally increases, as shown in Fig. 6.20. These results support the initial assumption that the Barlow Twins method benefits from the mixed-sampling strategy, even though the method contrasts feature vector components and not the image views per se. Note that the retrieval performance is highest with 256 clusters and not with 512. The high performance with 256 clusters might be caused by an implicit preference of the Barlow Twins for smaller batch sizes, or it indicates that the hardness/variability of the Barlow Twins' batches requires finer tuning compared to SimCLR.

Applying the proposed sampling strategies to BYOL produces unexpected results. BYOL does not contrast different images against each other and should be unaffected by the proposed sampling strategy. The in-cluster sampling results shown

Figure 6.20: Barlow Twins mixed-cluster sampling results with c clusters.Figure 6.21: BYOL in-cluster sampling results with c clusters and the batch size in parenthesis.

in Fig. 6.21 indicate that even with the same batch size, the sampling procedure has an impact on the overall retrieval performance. Changes to the relationship among input patches within batches *should* have little to no effect on BYOL's performance since it only contrasts positive pairs. There is research indicating that the standard BYOL method implicitly contrasts samples within a batch due to the batch normalization layers [82]. The implicit contrasting of all images in a batch supports the results shown in Fig. 6.21.

The performance differences are even more significant for the mixed-cluster sampling strategy shown in Fig. 6.22. Further inspection has revealed that the different batch sizes cause the main performance differences. Fig. 6.23 compares the default sampling method with different batch sizes against the mixed-cluster sampling strategy. The BYOL results become unstable for small batch sizes due

6.2 Analysis of Proposed Metadata-Guided Sampling Framework for CSSRL

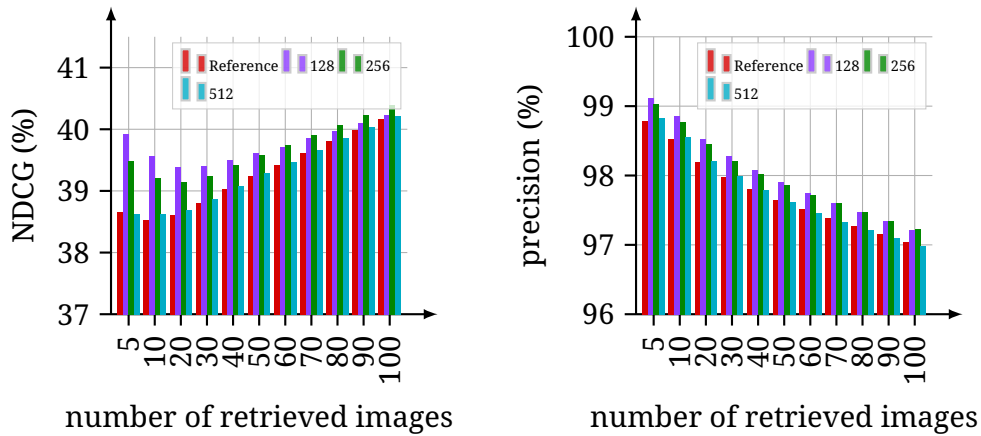


Figure 6.22: BYOL mixed-cluster sampling results with c clusters.

to the unstable batch statistics [83]. However, even when considering the different batch sizes, there is still a noticeable performance difference caused by the proposed mixed-cluster sampling strategy.

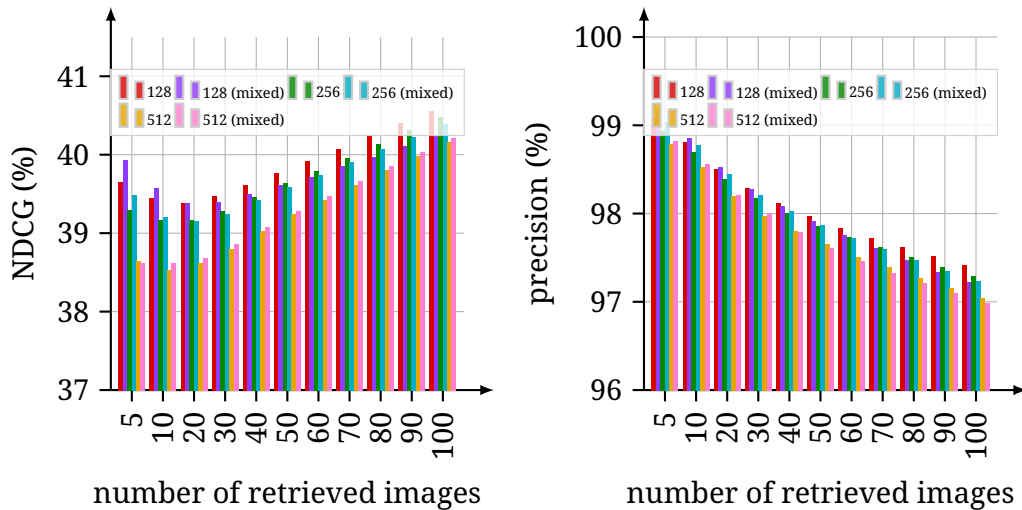


Figure 6.23: BYOL default sampling with different batch sizes vs. mixed-cluster sampling results with c clusters.

Recent research has shown that BYOL can be modified to work without a batch normalization layer with minor performance loss [84]. Future experiments could investigate if a non-batch normalization BYOL variant produces similar results with different sampling strategies. If the performance would still differ without batch normalization, it would indicate that BYOL is sensitive to over-/undersampling subsets. In the in-cluster sampling scenario, similar positive pairs might result in a uniform gradient direction that pushes the model towards early local minima. On the other hand, the mixed-cluster sampling scenario could have

the opposite effect and implicitly favor exploration of the loss-space due to the non-uniform gradient direction.

The presented CSSRL methods unanimously favor the mixed-cluster sampling strategy and, therefore, easier, more variable batches. The in-cluster sampling strategy hurts the retrieval performance in all scenarios and worsens with increasingly harder batches. These results indicate that the location metadata can effectively tune the batch hardness. The proposed sampling framework can be used to improve the overall image retrieval performance of the presented CSSRL methods.

7 Conclusion

The thesis proposes a general metadata-guided sampling framework to improve the content-based image retrieval performance of contrastive self-supervised representation learning (CSSRL) methods in the remote sensing (RS) domain. The sampling framework is applied to the SimCLR, Barlow Twins, and BYOL methods. The specific implementation clusters the dataset based on the location data. These clusters are then used to assemble batches with variable hardness as inputs for the different CSSRL methods. Tuning the batch hardness improves the general retrieval performance of these methods. Furthermore, the default augmentation pipeline from the natural image domain is critically evaluated. An augmentation pipeline tailored explicitly for the remote sensing domain is presented and experimentally verified.

The main results of investigating the default augmentation pipeline from the natural image domain in the RS setting are the following: Allowing small crops for the resized cropping augmentation is crucial for the retrieval performance of CSSRL methods, even though the probability of dropping entire classes from the input images is high in the RS domain. Small crops allow the model to learn scale-invariant features better and avoid short-circuit solutions, such as histogram matching. Dihedral and rotation transformations should be included since they create an inductive bias to learn rotation-invariant visual features. Rotational invariance is a unique property of remote sensing imagery and is desired for almost all remote sensing retrieval systems. Gaussian blurring can be included but should be evaluated depending on the specific method and the underlying dataset. Color-jittering should not be included in the augmentation pipeline since the main benefit of multi-spectral imagery is the ability to distinguish objects based on their spectral reflectivity. Although color-shifting is recommended in the natural image domain, the CSSRL methods' retrieval performance suffers from changes to the spectral reflectance values in the RS domain. As a result, the proposed augmentation pipeline consists of resized cropping, rotating, Dihedral transformation, and Gaussian blurring. The pipeline does *not* include any color-shifting transformations.

The first proposed sampling strategy, in-cluster sampling, generally increases the batch hardness by sampling all patches from a single cluster. The second sampling strategy, mixed-cluster sampling, decrease the hardness by sampling all patches from different clusters. The SimCLR and Barlow Twins method contrast all images within a batch and benefit from the sampling-based batch tuning approach. Increasing the batch hardness by reducing the input variability through in-cluster sampling hurts the overall retrieval performance of both methods. Increasing

the variability with the mixed-cluster sampling strategy generally improves the performance. These results show that the freely available location data can be used as a proxy for the patch similarity and implicitly used to tune the batch hardness.

BYOL does not contrast all images within a batch and should be mostly unaffected by the proposed augmentation strategy. However, the results are similar to those from the SimCLR and Barlow Twins experiments. The in-cluster sampling strategy decreases the retrieval performance, while the mixed-cluster sampling strategy slightly improves the results for the default configuration. The performance impact could be caused by implicitly contrasting all images within a batch due to the batch normalization layer, or it could be caused by oversampling sparse regions. The exact reason could be investigated in a future extension of this work.

Since the CSSRL methods do not require any labels, an exciting investigation could be to pre-train the models on a larger dataset and evaluate the performance on the BigEarthNet archive. Increasing the dataset size allows the CSSRL methods to learn more general features and imposes fewer restrictions on the sampling framework.

Future experiments could investigate dynamic sampling strategies. For example, the number of clusters to sample from could be modified during training. For the in-cluster sampling strategy, starting with few easy clusters and gradually increasing the hardness might increase the convergence speed or improve the model's overall performance.

Another possible extension to the proposed metadata-guided sampling framework is to pre-group the data based on temporal information. Embedding temporal information into the clustering pipeline would better accommodate for seasonal changes and create finer clusters.

The clusters themselves could also be dynamically updated in an online fashion. The location-based clustering result would be used as an *initial guess* for similar images. During training, the images' feature vectors could be compared against each other and, based on some statistics, be re-assigned to a different cluster.

The sampling framework could also be used to cluster the labels in hamming space and run similar experiments in a supervised fashion. These supervised experiments could be applied to the proposed CSSRL methods to investigate the batch dynamics further.

Bibliography

- [1] Famao Ye et al. “Remote Sensing Image Retrieval Using Convolutional Neural Network Features and Weighted Distance”. In: *IEEE Geosci. Remote Sensing Lett.* 15.10 (2018).
- [2] Xin-Yi Tong et al. “Exploiting Deep Features for Remote Sensing Image Retrieval: A Systematic Investigation”. In: *IEEE Trans. Big Data* 6.3 (July 23, 2017).
- [3] S.K. Sudha and S. Aji. “A Review on Recent Advances in Remote Sensing Image Retrieval Techniques”. In: *J Indian Soc Remote Sens* 47.12 (2019).
- [4] Jeremie Mouginot et al. “Comprehensive Annual Ice Sheet Velocity Mapping Using Landsat-8, Sentinel-1, and RADARSAT-2 Data”. In: *Remote Sensing* 9.4 (Apr. 2017), p. 364.
- [5] Aaron D. Gerace et al. “Increased potential to monitor water quality in the near-shore environment with Landsat’s next-generation satellite”. In: *Journal of Applied Remote Sensing* 7.1 (May 2013), p. 073558.
- [6] Jean-François Pekel et al. “High-resolution mapping of global surface water and its long-term changes”. In: *Nature* 540.7633 (Dec. 2016), pp. 418–422.
- [7] Yansheng Li et al. “Image retrieval from remote sensing big data: A survey”. In: *Information Fusion* 67 (Mar. 2021).
- [8] Michael A. Wulder et al. “Current status of Landsat program, science, and applications”. In: *Remote Sensing of Environment* 225 (May 2019), pp. 127–147.
- [9] Karen Fletcher. *Sentinel-2 : Esa’s optical high-resolution mission for GMES operational services*. ESA Communications, 2012.
- [10] Noel Gorelick et al. “Google Earth Engine: Planetary-scale geospatial analysis for everyone”. In: *Remote Sensing of Environment* 202 (Dec. 2017), pp. 18–27.
- [11] European Space Agency. *Copernicus Open Access Hub*. 2021. URL: <https://scihub.copernicus.eu/>.
- [12] Peter Baumann et al. “Big Data Analytics for Earth Sciences: the EarthServer approach”. In: *International Journal of Digital Earth* 9.1 (Mar. 2015), pp. 3–29.
- [13] Phuc H. Le-Khac et al. “Contrastive Representation Learning: A Framework and Review”. In: *#IEEE_O_ACC#* 8 (Oct. 10, 2020).

- [14] Robert M. Haralick et al. “Textural Features for Image Classification”. In: *IEEE Transactions on Systems, Man, and Cybernetics SMC-3.6* (Nov. 1973), pp. 610–621.
- [15] David G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. In: *International Journal of Computer Vision* 60.2 (Nov. 2004), pp. 91–110.
- [16] Grant J. Scott et al. “Entropy-Balanced Bitmap Tree for Shape-Based Object Retrieval From Large-Scale Satellite Imagery Databases”. In: *IEEE Transactions on Geoscience and Remote Sensing* 49.5 (May 2011), pp. 1603–1616.
- [17] T. Bretschneider et al. “Retrieval of remotely sensed imagery using spectral information content”. In: *IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2002.
- [18] Kenneth W. Tobin et al. “Large-Scale Geospatial Indexing for Image-Based Retrieval and Analysis”. In: *Advances in Visual Computing*. Springer Berlin Heidelberg, 2005, pp. 543–552.
- [19] Yikun Li and Timo R. Bretschneider. “Semantic-Sensitive Satellite Image Retrieval”. In: *IEEE Transactions on Geoscience and Remote Sensing* 45.4 (Apr. 2007), pp. 853–860.
- [20] Alex Krizhevsky et al. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*. 2012.
- [21] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2016.
- [22] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: (Sept. 2014). arXiv: 1409.1556 [cs.CV].
- [23] Matthew D. Zeiler and Rob Fergus. “Visualizing and Understanding Convolutional Networks”. In: *Computer Vision – ECCV 2014*. Springer International Publishing, 2014, pp. 818–833.
- [24] Yi Yang and Shawn Newsam. “Bag-of-visual-words and spatial extensions for land-use classification”. In: *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '10*. ACM Press, 2010.
- [25] Zhenfeng Shao et al. “Performance Evaluation of Single-Label and Multi-Label Remote Sensing Image Retrieval Using a Dense Labeling Dataset”. In: *Remote Sensing* 10.6 (June 2018), p. 964.

-
- [26] Gencer Sümbül et al. “Bigearthnet: A Large-Scale Benchmark Archive for Remote Sensing Image Understanding”. In: *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2019.
- [27] Timo Milbich et al. “DiVA: Diverse Visual Feature Aggregation for Deep Metric Learning”. In: (Apr. 28, 2020).
- [28] Peng Li and Peng Ren. “Partial Randomness Hashing for Large-Scale Remote Sensing Image Retrieval”. In: *IEEE Geoscience and Remote Sensing Letters* 14.3 (Mar. 2017).
- [29] Weiwei Song et al. “Deep Hashing Learning for Visual and Semantic Retrieval of Remote Sensing Images”. In: *IEEE Trans. Geosci. Remote Sensing* (2020). arXiv: 1909.04614 [cs.CV].
- [30] Min-Sub Yun et al. “Coarse-to-Fine Deep Metric Learning for Remote Sensing Image Retrieval”. In: *Remote Sensing* 12.2 (Oct. 19, 2020).
- [31] Hao Su et al. “Crowdsourcing Annotations for Visual Object Detection”. In: (2012).
- [32] Olga Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge”. In: *Int J Comput Vis* 115.3 (2015).
- [33] Bindita Chaudhuri et al. “Multilabel Remote Sensing Image Retrieval Using a Semisupervised Graph-Theoretic Method”. In: *IEEE Trans. Geosci. Remote Sensing* 56.2 (2018).
- [34] Giannis Karamanolakis et al. “Leveraging Just a Few Keywords for Fine-Grained Aspect Detection Through Weakly Supervised Co-Training”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, 2019.
- [35] Jinhui Tang et al. “Inferring semantic concepts from community-contributed images and noisy tags”. In: *Proceedings of the seventeen ACM international conference on Multimedia - MM '09*. ACM Press, 2009.
- [36] Carl Doersch et al. “Unsupervised Visual Representation Learning by Context Prediction”. In: *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. IEEE Computer Society, 2015.
- [37] Mathilde Caron et al. “Deep Clustering for Unsupervised Learning of Visual Features”. In: *Computer Vision – ECCV 2018* (July 15, 2018).
- [38] Mathilde Caron et al. “Unsupervised Pre-Training of Image Features on Non-Curated Data”. In: *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 2019.

- [39] Spyros Gidaris et al. “Unsupervised Representation Learning by Predicting Image Rotations”. In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.
- [40] Kaiming He et al. “Momentum Contrast for Unsupervised Visual Representation Learning”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. IEEE, 2020.
- [41] Ting Chen et al. “A Simple Framework for Contrastive Learning of Visual Representations”. In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020.
- [42] Mathilde Caron et al. “Unsupervised Learning of Visual Features by Contrasting Cluster Assignments”. In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. 2020.
- [43] Jean-Bastien Grill et al. “Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 21271–21284.
- [44] Jure Zbontar et al. “Barlow Twins: Self-Supervised Learning via Redundancy Reduction”. In: (Mar. 2021). arXiv: 2103.03230 [cs.CV].
- [45] Mathilde Caron et al. “Emerging Properties in Self-Supervised Vision Transformers”. In: (Apr. 2021). arXiv: 2104.14294 [cs.CV].
- [46] Longlong Jing and Yingli Tian. “Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* (2020).
- [47] John McCormac et al. “SceneNet RGB-D: Can 5M Synthetic Images Beat Generic ImageNet Pre-training on Indoor Segmentation?” In: *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2017.
- [48] G. E. Hinton. “Reducing the Dimensionality of Data with Neural Networks”. In: *Science* 313.5786 (July 2006), pp. 504–507.
- [49] Ian Goodfellow et al. “Generative Adversarial Nets”. In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc., 2014.
- [50] Deepak Pathak et al. “Context Encoders: Feature Learning by Inpainting”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016.
- [51] Richard Zhang et al. “Colorful Image Colorization”. In: (Mar. 2016). arXiv: 1603.08511 [cs.CV].

- [52] Mehdi Noroozi and Paolo Favaro. “Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles”. In: *ECCV* (Mar. 30, 2016).
- [53] Florian Schroff et al. “FaceNet: A unified embedding for face recognition and clustering”. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, 2015.
- [54] Zhuang Ma and Michael Collins. “Noise Contrastive Estimation and Negative Sampling for Conditional Models: Consistency and Statistical Efficiency”. In: *EMNLP*. 2018, pp. 3698–3707.
- [55] R. Manmatha et al. “Sampling Matters in Deep Embedding Learning”. In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 2017.
- [56] Ben Harwood et al. “Smart Mining for Deep Metric Learning”. In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 2017.
- [57] W. R. Tobler. “A Computer Movie Simulating Urban Growth in the Detroit Region”. In: *Economic Geography* 46 (June 1970), p. 234.
- [58] Oscar Mañas et al. “Seasonal Contrast: Unsupervised Pre-Training from Uncurated Remote Sensing Data”. In: (Mar. 2021). arXiv: 2103.16607 [cs.CV].
- [59] Neal Jean et al. “Tile2Vec: Unsupervised Representation Learning for Spatially Distributed Data”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33 (July 2019), pp. 3967–3974.
- [60] Jian Kang et al. “Deep Unsupervised Embedding for Remotely Sensed Images Based on Spatially Augmented Momentum Contrast”. In: *IEEE Transactions on Geoscience and Remote Sensing* 59.3 (Mar. 2021), pp. 2598–2610.
- [61] Kumar Ayush et al. “Geography-Aware Self-Supervised Learning”. In: (Nov. 2020). arXiv: 2011.09980 [cs.CV].
- [62] Stuart Lloyd. “Least squares quantization in PCM”. In: *IEEE Transactions on Information Theory* 28.2 (Mar. 1982), pp. 129–137.
- [63] Leonard Kaufman and Peter J. Rousseeuw. “Partitioning Around Medoids (Program PAM)”. In: *Finding Groups in Data*. John Wiley & Sons, Inc., 1990, pp. 68–125.
- [64] Erich Schubert and Peter J. Rousseeuw. “Faster k-Medoids Clustering: Improving the PAM, CLARA, and CLARANS Algorithms”. In: *Similarity Search and Applications*. Springer International Publishing, 2019, pp. 171–187.
- [65] Erich Schubert and Peter J. Rousseeuw. “Fast and eager k-medoids clustering: O(k) runtime improvement of the PAM, CLARA, and CLARANS algorithms”. In: *Information Systems* 101 (Nov. 2021), p. 101804.
- [66] European Space Agency. *Cloud Masks*. June 2021. URL: <https://sentinel.esa.int/web/sentinel/technical-guides/sentinel-2-msi/level-1c/cloud-masks>.

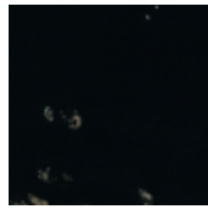
- [67] European Space Agency. *Heritage*. 2021. URL: <https://sentinel.esa.int/web/sentinel/missions/sentinel-2/heritage>.
- [68] Yingqi Zhang et al. “Attention-Aware Joint Location Constraint Hashing for Multi-Label Image Retrieval”. In: *IEEE Access* 8 (2020).
- [69] Zheng Zhang et al. “Improved Deep Hashing With Soft Pairwise Similarity for Multi-Label Image Retrieval”. In: *IEEE Transactions on Multimedia* 22.2 (Feb. 2020).
- [70] Kalervo Järvelin and Jaana Kekäläinen. “IR evaluation methods for retrieving highly relevant documents”. In: *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '00*. ACM Press, 2000.
- [71] Kalervo Järvelin and Jaana Kekäläinen. “Cumulated gain-based evaluation of IR techniques”. In: *ACM Transactions on Information Systems* 20.4 (Oct. 2002).
- [72] Richard Zhang et al. “Real-time user-guided image colorization with learned deep priors”. In: *ACM Trans. Graph.* 36.4 (Mar. 28, 2016).
- [73] Dask Development Team. *Dask: Library for dynamic task scheduling*. 2016.
- [74] Saining Xie et al. “Aggregated Residual Transformations for Deep Neural Networks”. In: (Nov. 2016). arXiv: 1611.05431 [cs.CV].
- [75] Jie Hu et al. “Squeeze-and-Excitation Networks”. In: (Sept. 2017). arXiv: 1709.01507 [cs.CV].
- [76] Tong He et al. “Bag of Tricks for Image Classification with Convolutional Neural Networks”. In: (Dec. 2018). arXiv: 1812.01187 [cs.CV].
- [77] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: (Dec. 2014). arXiv: 1412.6980 [cs.LG].
- [78] Jeremy Howard and Sylvain Gugger. “fastai: A Layered API for Deep Learning”. In: *Information* 2020, 11(2), 108 (Feb. 2020). arXiv: 2002.04688 [cs.LG].
- [79] Kerem Turgutlu et al. *KeremTurgutlu/self_supervised: DINO*. 2021.
- [80] Gencer Sümbül and Begüm Demir. “A Novel Graph-Theoretic Deep Representation Learning Method for Multi-Label Remote Sensing Image Retrieval”. In: (June 2021). arXiv: 2106.00506 [cs.CV].
- [81] Less Wright and Nestor Demeure. “Ranger21: a synergistic deep learning optimizer”. In: (June 2021). arXiv: 2106.13731 [cs.LG].
- [82] Abe Fetterman and Josh Albrecht. *Understanding self-supervised and contrastive learning with bootstrap your own latent (BYOL)*. 2020. URL: <https://generallyintelligent.ai/understanding-self-supervised-contrastive-learning.html>.

- [83] Zeming Li et al. “Momentum² Teacher: Momentum Teacher with Momentum Statistics for Self-Supervised Learning”. In: (Jan. 2021). arXiv: 2101.07525 [cs.LG].
- [84] Pierre H. Richemond et al. “BYOL works even without batch statistics”. In: (Oct. 2020). arXiv: 2010.10241 [stat.ML].

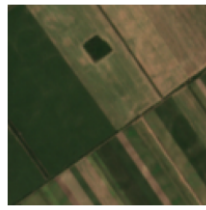
A Additional BigEarthNet Resources

Table A.1: Relation between New and Original Class-Nomenclature.

Recommended 19-label class-nomenclature	Associated original 43 class-nomenclature
Urban fabric	Continuous urban fabric; Discontinuous urban fabric
Industrial or commercial units	Industrial or commercial units
Arable land	Non-irrigated arable land; Permanently irrigated land; Rice fields
Permanent crops	Vineyards; Fruit trees and berry plantations; Olive groves; Annual crops associated with permanent crops
Pastures	Pastures
Agro-forestry areas	Agro-forestry areas
Complex cultivation patterns	Complex cultivation patterns
Broad-leaved forest	Broad-leaved forest
Coniferous forest	Coniferous forest
Mixed forest	Mixed forest
Natural grassland and sparsely vegetated areas	Natural grassland; Sparsely vegetated areas
Moors, heathland and sclerophyllous vegetation	Moors and heathland; Sclerophyllous vegetation
Beaches, dunes, sands	Beaches, dunes, sands
Transitional woodland, shrub	Transitional woodland/shrub
Inland wetlands	Inland marshes; Peatbogs
Coastal wetlands	Salt marshes; Salines
Inland waters	Water courses; Water bodies
Marine waters	Coastal lagoons; Estuaries; Sea and ocean
Land principally occupied by agriculture, with significant areas of natural vegetation	Land principally occupied by agriculture, with significant areas of natural vegetation
-	Airports
-	Bare rock
-	Dump sites
-	Port areas
-	Road and rail networks and associated land
-	Mineral extraction sites
-	Construction sites
-	Sport and leisure facilities
-	Burnt areas
-	Intertidal flats
-	Green urban areas



(a) Marine waters



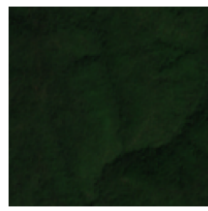
(b) Arable land



(c) Inland waters



(d) Pastures



(e) Broad-leaved forest



(f) Agro-forestry areas



(g) Coniferous forest



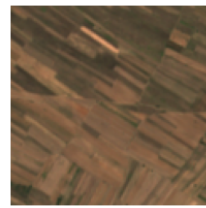
(h) Inland wetlands



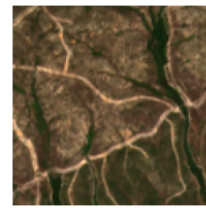
(i) Mixed forest



(j) Moors
heathland and
sclerophyllous
vegetation



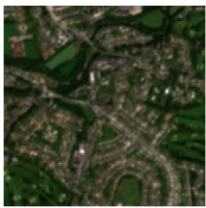
(k) Complex
cultivation
patterns



(l) Transitional
woodland
shrub



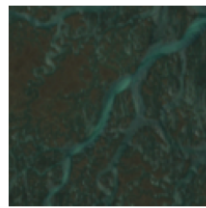
(m) Permanent
crops



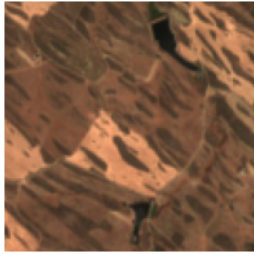
(n) Urban fabric



(o) Natural
grassland and
sparsely
vegetated
areas



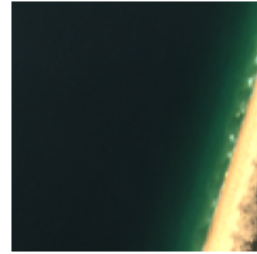
(p) Coastal
wetlands



(q) Land principally occupied by agriculture with significant areas of natural vegetation



(r) Industrial or commercial units



(s) Beaches dunes sands

Figure A.1: Example images containing the respective classes from the 19 class-nomenclature.

B Extended Experimental Results

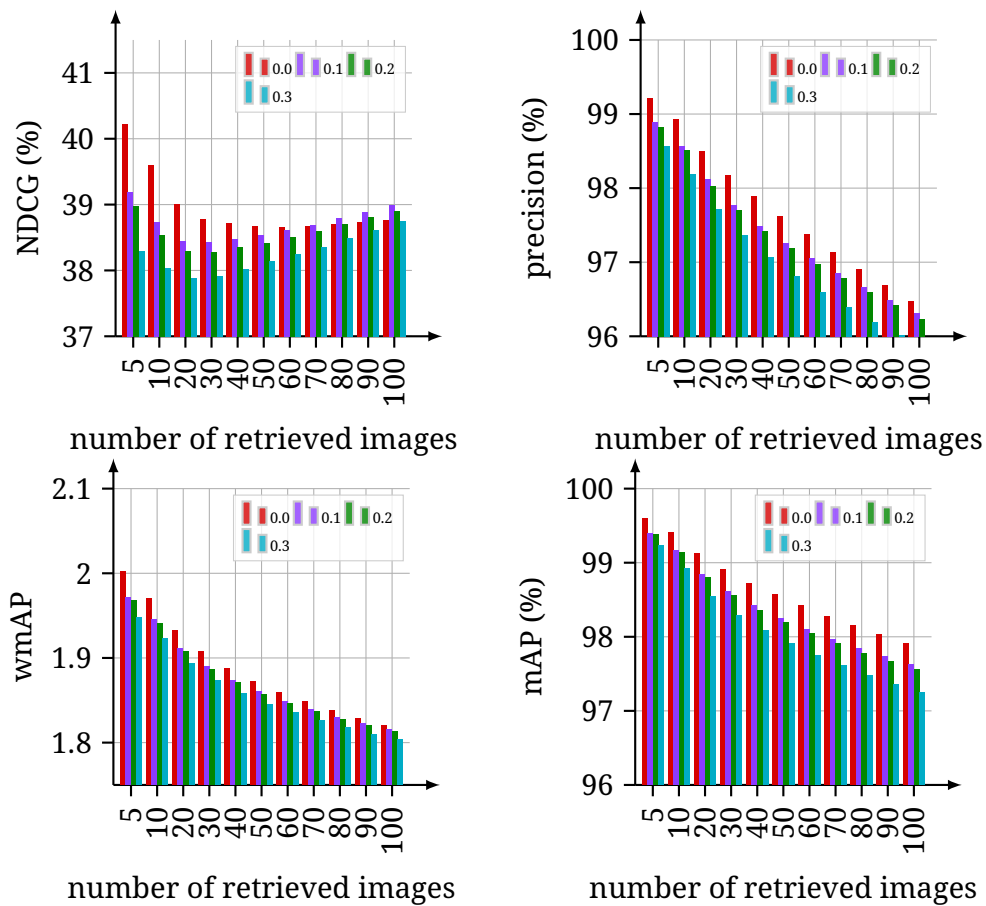


Figure B.1: Simplified multi-spectral color-jittering augmentation (brightness + contrast shifting) results with SimCLR method.

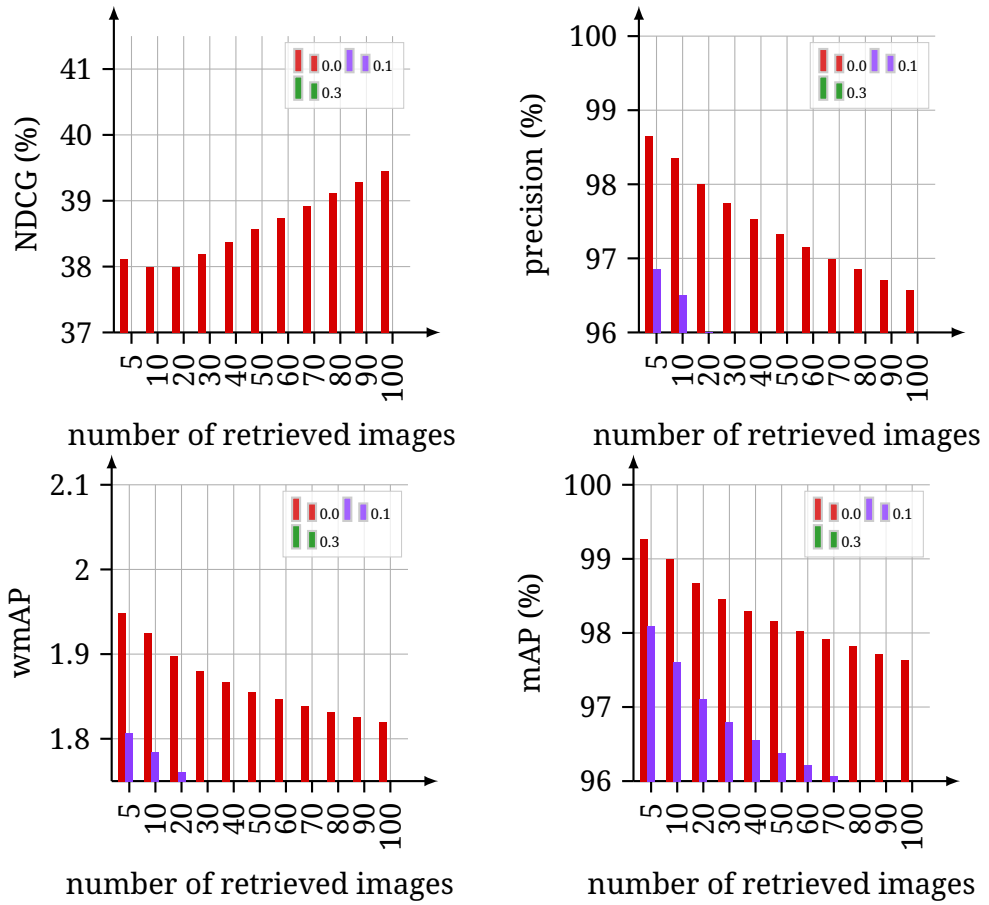


Figure B.2: Simplified multi-spectral color-jittering augmentation (brightness + contrast shifting) results with Barlow Twins method.

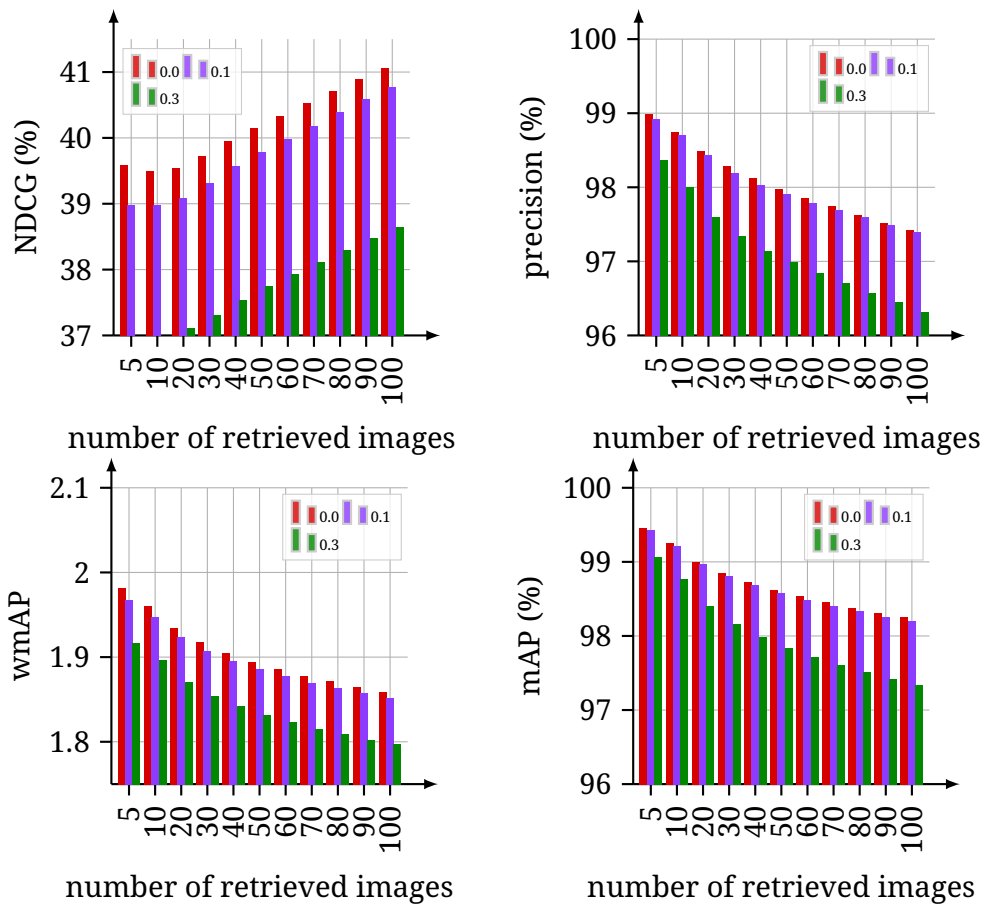


Figure B.3: Simplified multi-spectral color-jittering augmentation (brightness + contrast shifting) results with BYOL method.

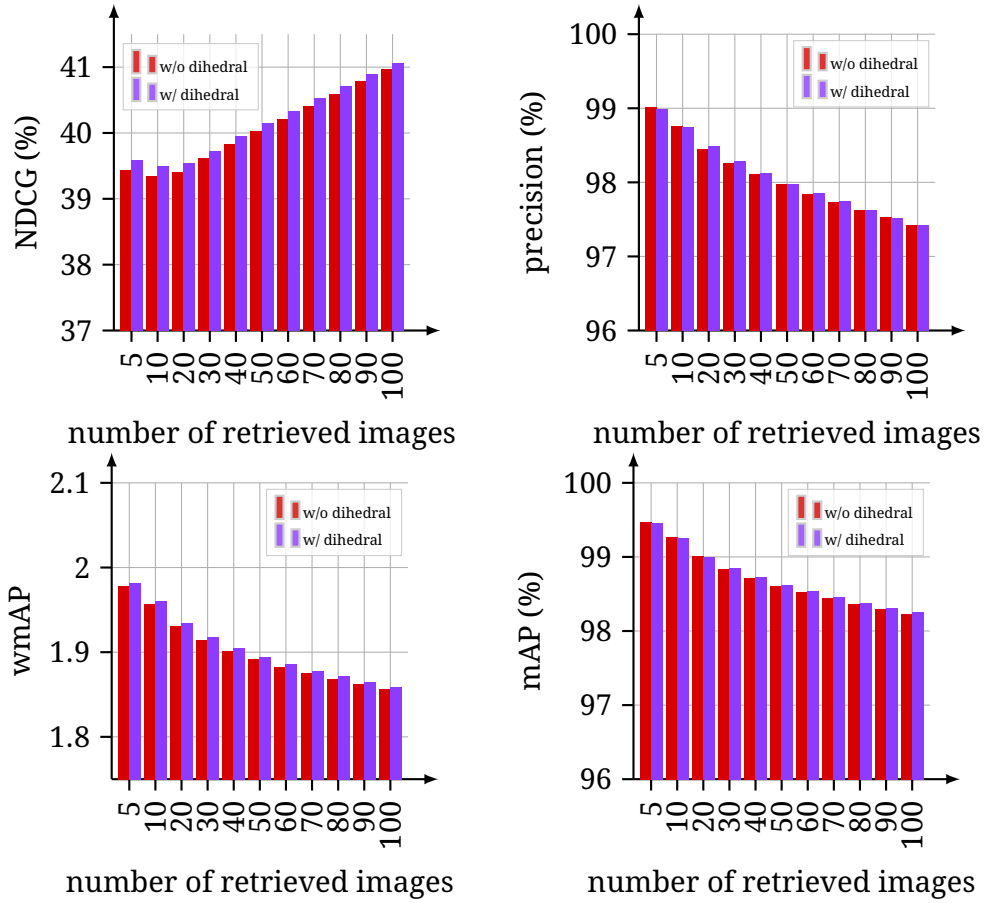


Figure B.4: Dihedral transformation results with BYOL method.

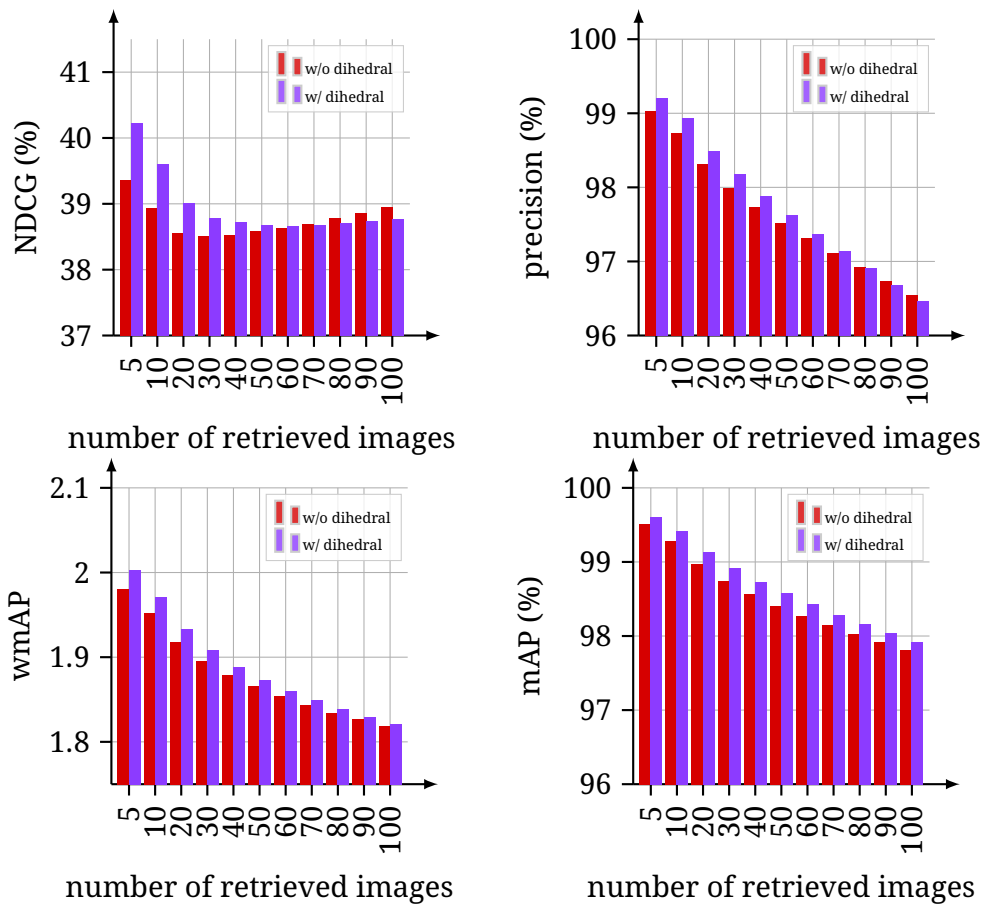


Figure B.5: Dihedral transformation results with SimCLR method.

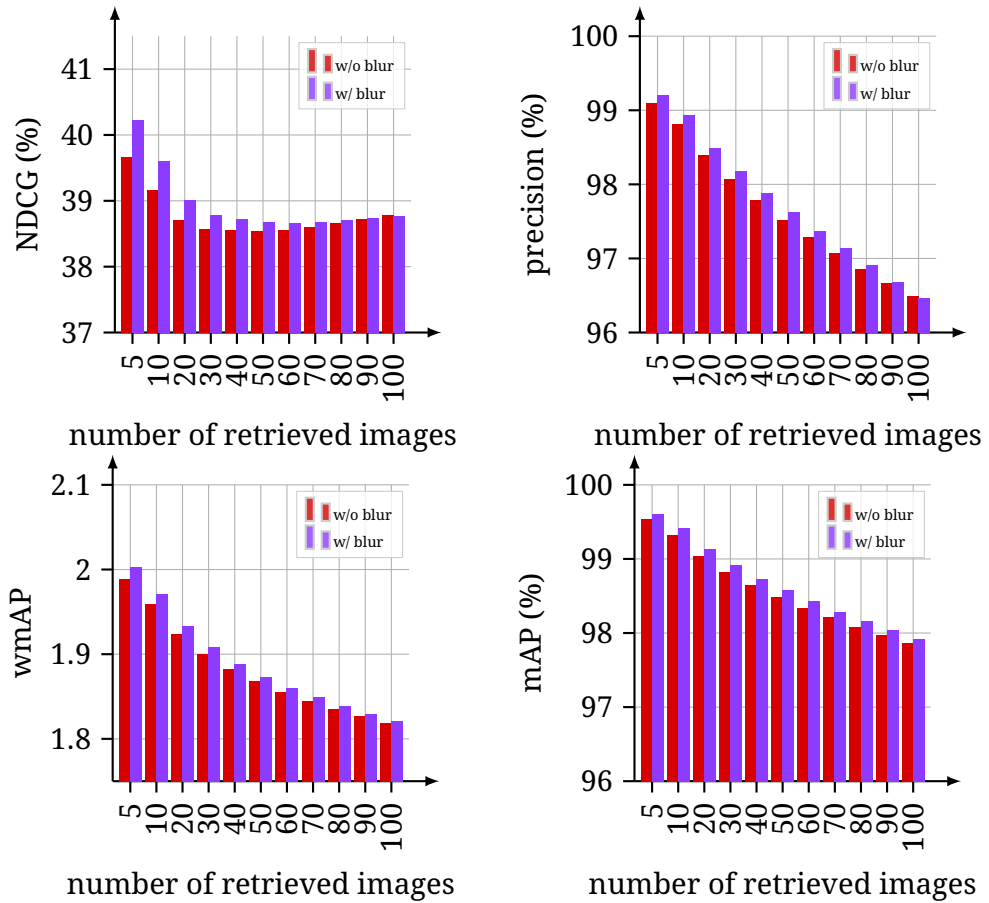


Figure B.6: Gaussian Blurring results with SimCLR method.

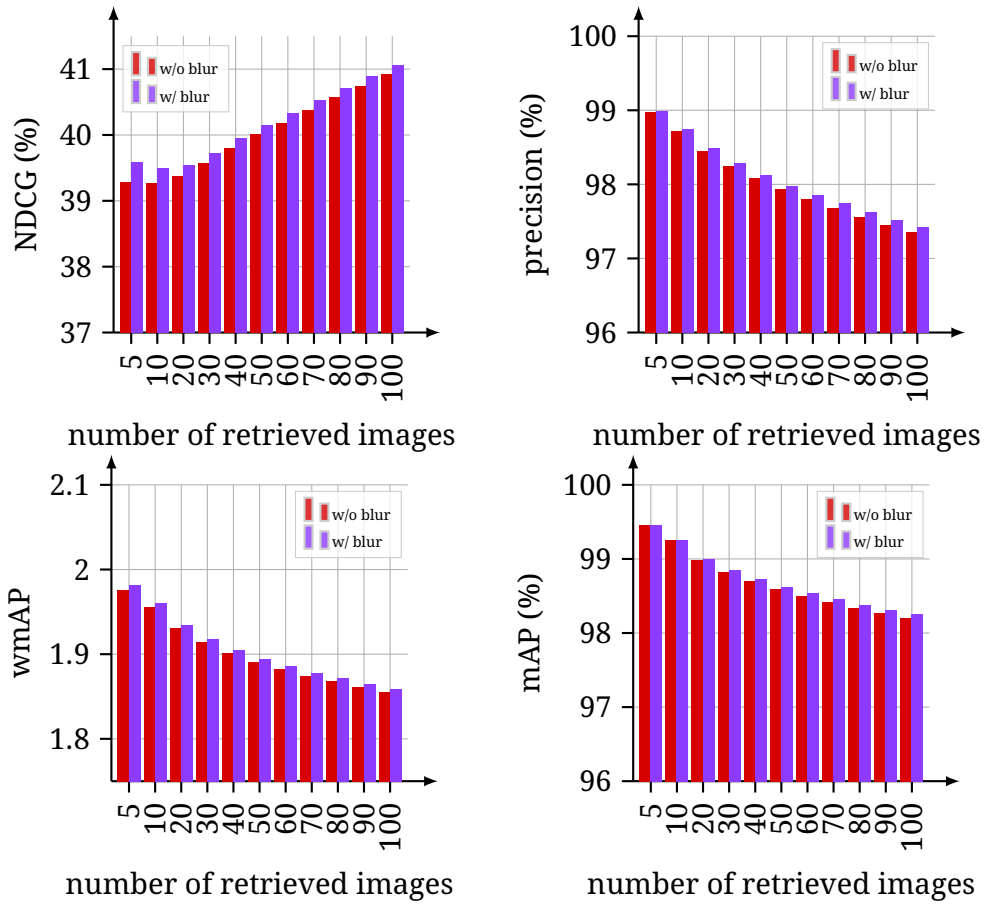


Figure B.7: Gaussian Blurring results with BYOL method.

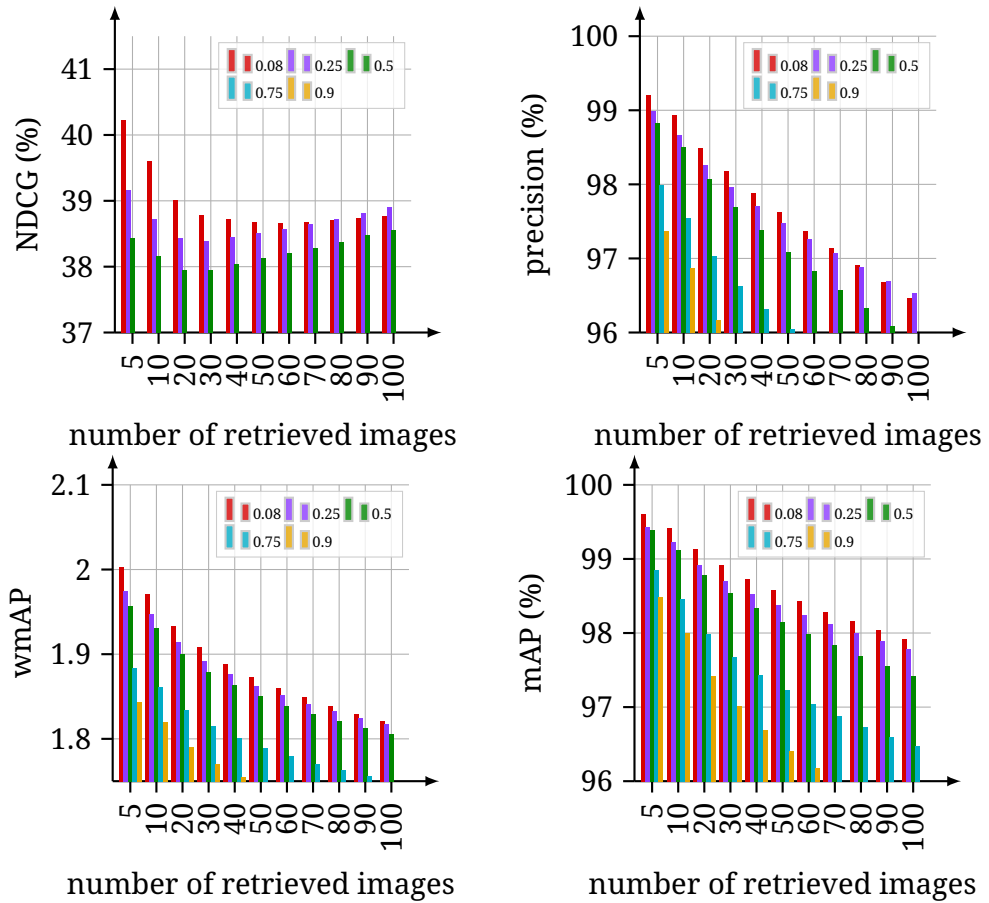


Figure B.8: Resized cropping results with SimCLR method.

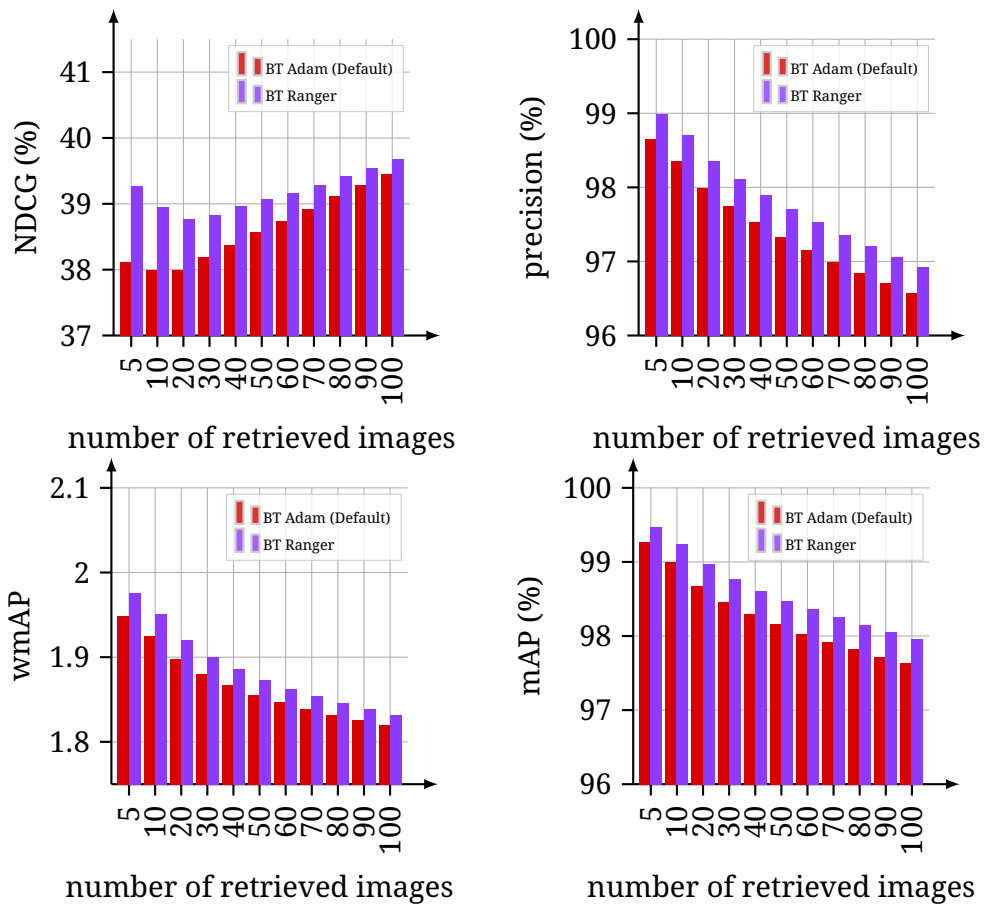


Figure B.9: Effect of Adam (default) vs. Ranger21 optimizer on Barlow Twins.

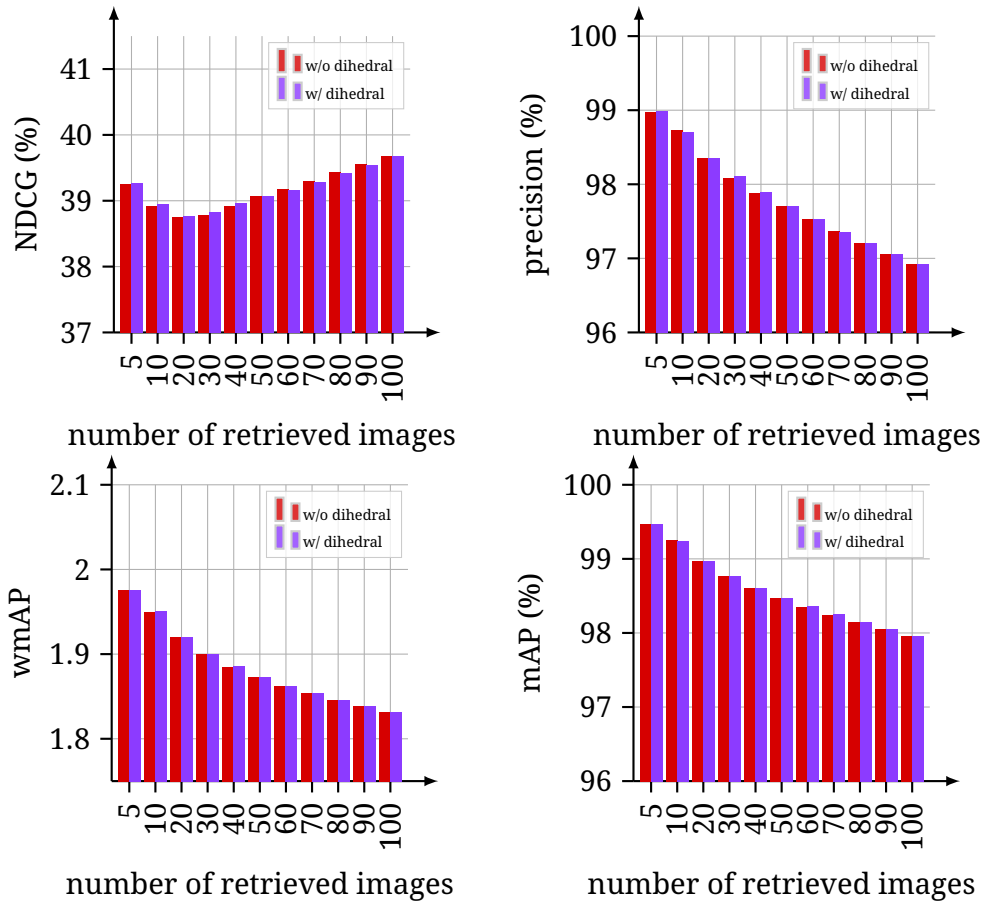


Figure B.10: Dihedral transformation results with Barlow Twins method + Ranger21 optimizer.

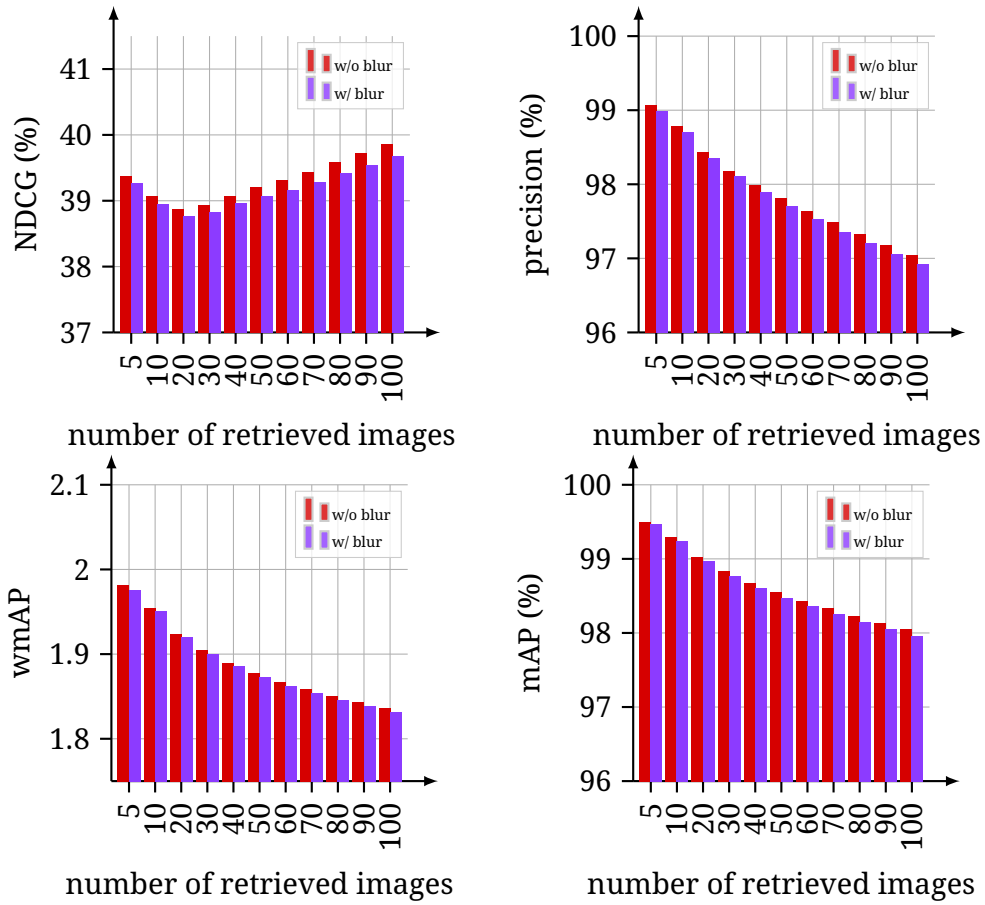


Figure B.11: Gaussian Blurring results with Barlow Twins method + Ranger21 optimizer.

B Extended Experimental Results

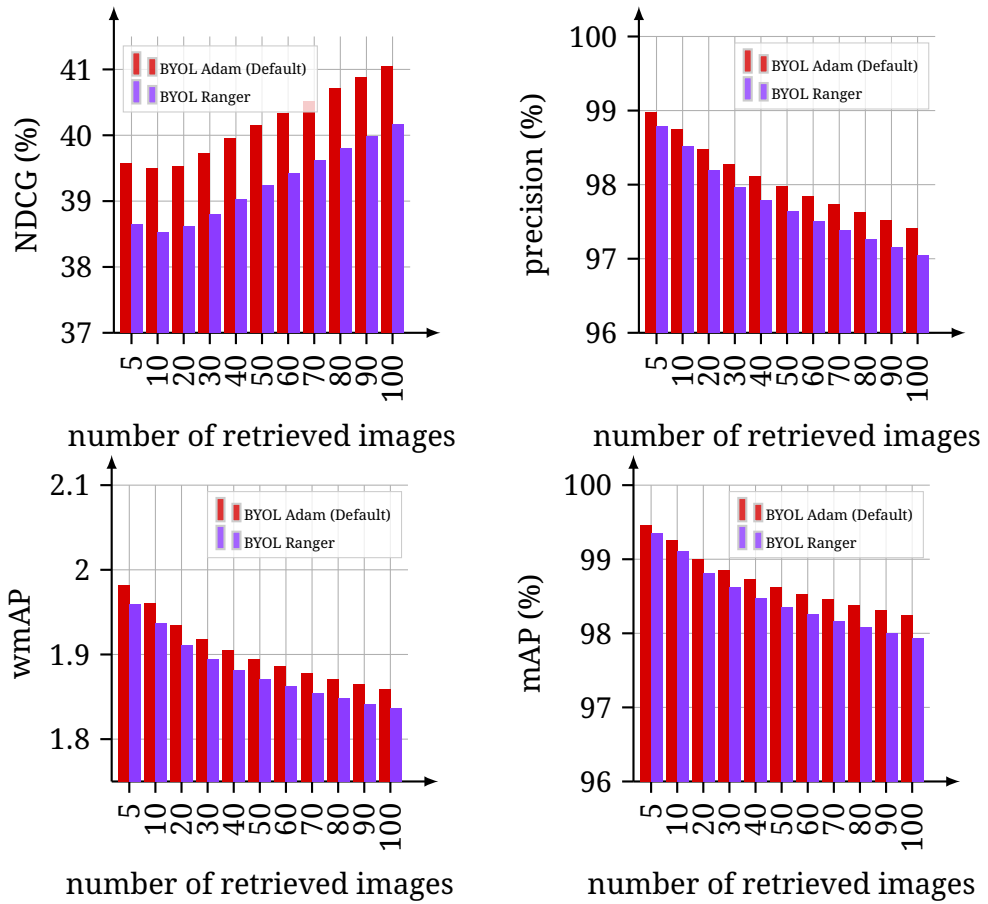


Figure B.12: Effect of Adam (default) vs. Ranger21 optimizer on BYOL.

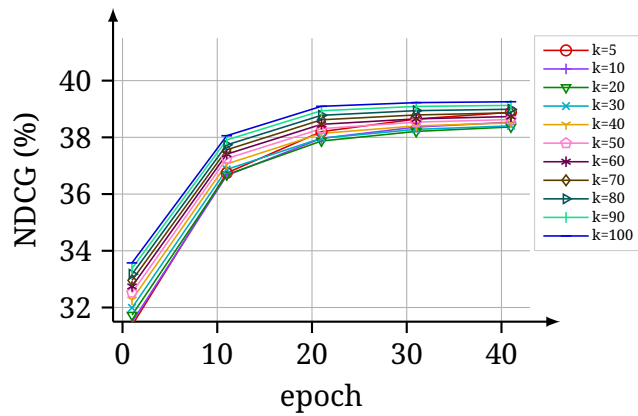


Figure B.13: SimCLR + Ranger21 + in-cluster sampling with 10 clusters. Displaying NDCG scores over time.