Semi and Self-Supervised Metric Learning for Remote Sensing Applications

Itza Hernandez-Sequeira, Ruben Fernandez-Beltran, *Senior Member, IEEE,* Filiberto Pla, *Senior Member, IEEE*

***Abstract*—Earth data collection from satellites and aircraft has exponentially grown, but a substantial portion of it remains unlabeled. This has prompted the remote sensing community to explore effective methods for leveraging unlabeled data. In our prior investigation [1], we evaluated various deep semi-supervised learning algorithms on two very high-resolution (VHR) optical datasets (UCM [2] and AID [3]). Notably, the CoMatch [4] algorithm demonstrated the highest accuracy, motivating further exploration. This letter extends our earlier work by integrating the established Class-Aware Contrastive Semi-Supervised Learning framework (Comatch+CCSSL) [5] into CoMatch and introducing a new triplet metric learning loss (CoMatch+Triplet). CoMatch+Triplet excelled with 93.2% accuracy on UCM, while CoMatch led with 92.19% on AID. The addition of the triplet loss can produce a clearer separation of the samples from different classes in the embedding space at very early learning stages, being able to learn faster and getting maximum performance with few iterations. The exploration of diverse semi and self-supervised training methodologies presented in this work sheds light on the strengths and limitations of these approaches, enhancing our understanding of their applicability in remote sensing applications.**

***Index Terms*—Semi-Supervised, Self-Supervised, Remote Sensing**

I. Introduction

THE vast volume of available remote sensing data, coupled with the cost and time constraints associated with obtaining labeled data in this domain, has sparked a growing interest in leveraging unlabeled data. Unlike traditional supervised approaches that demand a substantial amount of labeled data, unsupervised learning operates without any labeled data, semi-supervised learning effectively utilizes both labeled and unlabeled data and self-supervised learning generates labeled data on its own [6]. These frameworks demonstrate the efficacy of incorporating unlabeled data in image classification tasks, employing diverse learning strategies, data augmentation techniques, and adequate loss functions. Notably, these loss functions are integral to metric learning, enabling the acquisition of meaningful visual representations through approaches like pair and triplet losses [7]. This letter focuses on exploring the use of unlabeled data in domains such as semi-supervised learning and self-supervised learning. We implement the CoMatch [4] method, which concurrently learns two representations of the training data, i.e., their class probabilities and low-dimensional embeddings. Additionally, we add to CoMatch the Class-Aware Contrastive Semi-Supervised Learning framework (CCSSL) for separating in and out of distribution data. Finally, we also introduce a triplet loss, a concept of recurring interest in the remote sensing community for these learning frameworks [8].

Therefore, the main contributions of this work can be summarised as follows:

1) Apply CoMatch, a state-of-the-art algorithm and the CCSSL framework in remote sensing applications.
2) Experiment with metric learning and adding a triplet loss to contrastive learning approches.
3) Establish experimental protocols using very high-resolution (VHR) datasets UCM and AID to evaluate semi- and self-supervised learning algorithms in RS.

II. Related Work

In this section, we investigate the intersection of semi- and self-supervised learning. Then, we explore pair and triplet metric learning in the context of remote sensing applications.

A. Deep Semi-supervised and Self-supervised Learning

Semi-supervised learning algorithms are designed to leverage both labeled and unlabeled samples. This typically involves employing a standard labeled dataset, retaining only a fraction of the labels (e.g., 10%), and using the remaining data as unlabeled [9]. Within this framework, FixMatch [10] is a notable algorithm that combines consistency regularization and pseudo-labeling. It generates artificial labels from weakly augmented images and enforces loss against strongly augmented versions. However, a drawback of FixMatch is its sole reliance on the model output for predictions, potentially leading to confirmation bias.

In contrast, self-supervised learning techniques utilize unlabeled data for representation learning. These methods train neural networks by performing pretext tasks, such as contrastive learning [11]. Notable examples include MoCo [12] using InfoNCE [13] and SimCLR using NT-Xent [14]. Following this initial phase, the pre-trained network is fine-tuned for downstream tasks, including image classification on labeled data [6].

Looking ahead, the domains of semi- and self-supervised learning complement each other. To overcome the limitations of semi-supervised learning, CoMatch [4] builds upon the FixMatch framework, incorporating innovative ideas from contrastive and graph-based learning. It employs a co-training framework where class probabilities and low-dimensional embeddings interact and co-evolve. Self-supervised visual representation is inherently class-agnostic, hence integrating a small number of labeled examples can improve its effectiveness.

This research was supported by the Ministry of Science and Innovation of Spain (PID 2021-128794OB-I00), and the University Jaume I (PREDOC/2020/50).

Manuscript received December 29, 2023; revised February 19, 2024.

B. Deep Metric Learning and Remote Sensing

Metric learning aims to place similar points close together and dissimilar points apart in a feature space. This is achieved through specific loss functions, like pair and triplet losses, which encourage the creation of well-clustered spaces [7]. The classic contrastive loss [11], attempts to minimize the distance between positive pairs, and maximize the distance between negative pairs by a threshold [7]. The triplet loss [15] comprises a positive and a negative image sample relative to a reference called an anchor. The goal of the triplet loss is to minimize the distances between the anchor and positive samples compared to the distances between the anchor and negative sample by a margin [7].

The utilization of pair and triplet losses has proven successful in the context of remote sensing data. The Tile2Vec [8] algorithm incorporates the triplet loss, based on spatial neighborhood principles, where close geographic neighbors (anchor and positive) are expected to share similar representations than distant ones (anchor and negative). However, challenges arise in the triplet selection process as illustrated in figure 1, adapted from [16], where A is the anchor image, $(P1, P2)$ are positive images, and $(N1, N2)$ are negative images. Randomly selected triplets $(A, P1, N1)$ may already satisfy the margins, resulting in no model updates. On the other hand, triplets that incur in high loss value are called hard triplets $(A, P1, N2)$ and prompt significant model parameter updates in the embedding space. Opting for informative triplets contributes to faster convergence, accelerated learning, and reduced computational complexity during training [16].

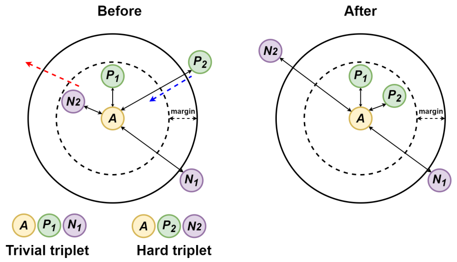

Fig. 1. Triplet Selection and Embedding Space Update (adapted from [16]). Here, positive images (P) are depicted moving closer to the anchor (A), as indicated by the blue arrow. In contrast, negative images (N) are pushed apart, as shown by the red arrow. A ”trivial triplet” (A, P1, N1) already satisfies the margins, whereas a ”hard triplet” (A, P2, N2) incurs significant updates.

To address triplet selection challenges, some remote sensing studies are exploring contrastive learning. [17] uses NT-Xent [14] with anchor and multiple neighbors, and [18] introduces Spatially Augmented Momentum Contrast (SauMoCo), utilizing Info-NCE [12].

III. Methodology

In this section, we extend the findings of [1] and explore three variations of the Comatch learning framework: CoMatch, CoMatch+CCSSL, and CoMatch+Triplet.

A. Dataset modifications and augmentation strategies

Our initial experiments were conducted on the UCM dataset [2], comprising 21 classes, each with 100 images of 256×256 pixels. Additionally, the AID dataset [3] was employed, featuring 30 classes with around 200 to 400 samples per class, and each sample measured 600×600 pixels. Some classes achieved significantly high accuracy, influencing the overall accuracy. Therefore, we conducted assessments under a constrained scenario, where both datasets were reduced to 10 classes (figure 2). To challenge the model, we selected classes with higher confusion rates based on the initial confusion matrix obtained from CoMatch with a ResNet18 backbone.

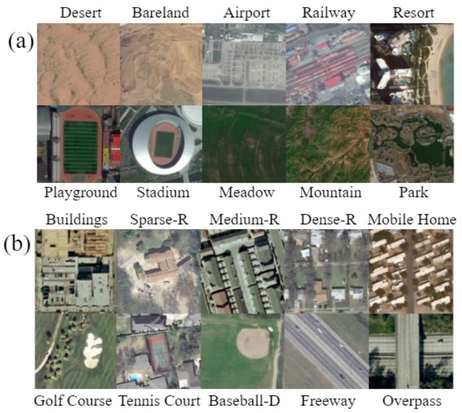

Fig. 2. Reduced datasets: (a) AID and (b) UCM with 10 classes with the highest overlap/confusion. R is Residential and D for Diamond.

In terms of data augmentation strategies, we refer to $Aug(\cdot)$ as a random transformation that maintains the image label. The CoMatch algorithm requires the use of one weak $Aug_w(\cdot)$ involving a random horizontal flip and two strong augmentations where $Aug_{s1}(.)$ employs RandAugment and $Aug_{s2}(\cdot)$ implements color jittering and grayscaling [4]. Departing from the previous approach [1], we modified the input images by resizing them to 64x64 and center cropping to 60. This adjustment allows for an increased batch size, from 8 to 32 considering our computational resources available. The aim is to efficiently load more images into memory simultaneously by reducing their size. Ensuring a sufficiently large batch size is crucial for having multiple samples in a minibatch, enhancing model robustness and generalization ability. In this work, we used 4 labeled samples per class, differing from prior trials (4, 25, 40), to provide insights into the model's performance with a more limited labeled dataset.

B. Learning framework and set up

We utilized the PyTorch implementation of Class Aware Contrastive Semi-Supervised Learning (CCSSL) [5]. The repository contains both CoMatch and CoMatch+CCSSL algorithms. Then, we extended CoMatch by incorporating a triplet loss via the PyTorch Metric Learning open-source library [19], denoted as CoMatch+Triplet.

In **CoMatch** [4], the deep encoder network $f(\cdot)$ produces a high-dimensional feature $f(x)$ from an input image x. The classification head $h(\cdot)$ outputs class probabilities, $p(y|x) = h(f(x))$, and the non-linear projection head $g(\cdot)$ transforms features into normalized low-dimensional embeddings $z(x) = g(f(x))$ (Fig. 3). CoMatch operates on both labeled $\mathcal{X}$ and unlabeled $\mathcal{U}$ data batches, optimizing three losses: supervised classification loss $\mathcal{L}x$, unsupervised classification loss $\mathcal{L}u^{cls}$, and graph-based contrastive loss $\mathcal{L}_u^{ctr}$.

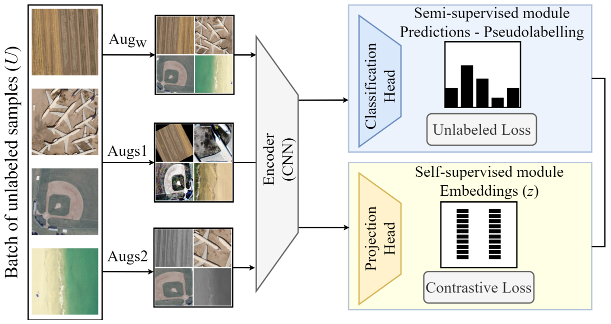

Fig. 3. Diagram of CoMatch (adapted from [4]).

For labeled data, $\mathcal{L}x$ uses cross-entropy between true labels and predictions with weak augmentations. The unsupervised classification loss $\mathcal{L}u^{cls}$ considers pseudo-labels and model predictions, employing strong augmentations. Pseudo-labels are retained based on a threshold τ without converting them to hard labels, differentiating from FixMatch. Entropy minimization is achieved through the contrastive loss $\mathcal{L}_u^{ctr}$ that uses cross-entropy between the pseudo-label graph W^q and the embedding graph W^z. The overall training objective is given by equation 1 where the hyperparameters λ_{cls} and λ_{ctr} control the weights.

$$\mathcal{L} = \mathcal{L}_x + \lambda_{cls}\mathcal{L}_u^{cls} + \lambda_{ctr}\mathcal{L}_u^{ctr} \tag{1}$$

For **CoMatch+CCSSL** [5], the Class-Aware Contrastive Semi-Supervised Learning (CCSSL) framework is added to CoMatch to enhance its capability to handle in-distribution and out-of-distribution data. CCSSL introduces class-aware contrastive learning, leveraging embeddings z from the same category as positive pairs using pseudo-labels q. For in-distribution data ($\max(p) > T_{push}$), class-aware clustering is applied, while for out-of-distribution data ($\max(p) < T_{push}$), a contrastive learning mechanism is employed. The overall training objective of CCSSL is a weighted sum of supervised loss $\mathcal{L}_x$, semi-supervised loss $\mathcal{L}_u$, and class-aware contrastive loss $\mathcal{L}_c$, with λ_u and λ_c as the weights.

Based on the ablation studies we configured CoMatch+CCSSL [5] for two scenarios (Table I). In out-distribution scenarios, the confidence threshold τ is lowered to 0.6 to allow more unlabeled samples to receive pseudo-labels. To address potential trade-offs, λ_c is increased from 1 to 2, augmenting the weight of the unsupervised contrastive loss. We activate the CCSSL module for high-noise datasets where the optimal T_{push} = 0.9. The labeled-to-unlabeled data ratio (μ) in a batch impacts CCSSL performance; we use a small ratio (μ = 5) when only using class-aware clustering without contrastive regularization (T_{push} = 0), and a larger ratio (μ = 7) when contrastive learning is increased.

In the **CoMatch+Triplet** variant, we emphasize enhancing the CoMatch framework by incorporating a triplet loss for improved embedding learning. Theoretically, by pulling similar samples together and dissimilar samples apart the model would reach a faster convergence. For the implementation, we used the open-source library of PyTorch Metric Learning [19]. In this approach, we treat the embeddings (z) derived from the initial set of strongly augmented unlabeled samples $\text{Aug}_{S1}(u)$ as our samples. The corresponding labels are the one-hot-encoded pseudo-labels obtained through the semi-supervised module $\max(q(\text{Aug}_{S1}(u)))$.

TABLE I
CONFIGURATION OF CCSSL PARAMETERS BASED ON NOISE LEVELS.

Scenario	**Threshold** (τ)	**Weight** (λ_c)	$T_{\textbf{push}}$	**Ratio** (μ)
In-distribution	0.95	1	0	5
Out-distribution	0.6	2	0.9	7

The pseudocode in III-B outlines the implementation of the TripletMarginLoss. Access the PyTorch implementation at https://github.com/itzahs/SSL-for-RS. In this setup, anchor-positive pairs consist of embeddings with the same label, while anchor-negative pairs consist of embeddings with different labels. We employ a TripletMarginMiner to dynamically select triplets in a semi-hard online mining approach. We used a cosine similarity instead of an Euclidean distance to prevent the loss of significance due to the curse of dimensionality [4]. The final loss is computed by adding the deep metric learning triplet loss to the CoMatch losses. The weighting factors, denoted as λ terms, are uniformly set to 1.

Pseudocode 1 CoMatch+Triplet loss

```
# This pseudocode is an extension of CoMatch.
Input: unlabeled samples $(u)$, encoder $f$, classifier $h$, projection head $g$, strongly-augmented embeddings $z = g \circ f(\text{Aug}_{S1}(u))$ and pseudo-labels $max(q(Aug_{S1}(u)))$.
Output: Combined $loss$
# Set margin, distance, triplet selection method.
$margin \leftarrow \texttt{0.2}$
$distance \leftarrow \texttt{CosineSimilarity()}$
$triplets \leftarrow \texttt{semihard}$
# Define deep metric learning loss.
$loss\_func \leftarrow \texttt{TripletMarginLoss}(margin, distance)$
# Set up loss in conjunction with a miner.
$miner \leftarrow \texttt{TripletMarginMiner}(margin, distance,$
                                        $triplets)$
$miner\_output \leftarrow \texttt{miner}(embeddings, labels)$
$loss_{dml} \leftarrow \texttt{loss\_func}(embeddings, labels, miner\_output)$
# Combine CoMatch loss with Triplet loss.
$loss \leftarrow \mathcal{L}x + \lambda cls\mathcal{L}u^{cls} + \lambda ctr\mathcal{L}u^{ctr} + \lambda dml\mathcal{L}_u^{dml}$
```

To ensure a fair comparison, we adhere to a set of constant parameters, including the choice of image augmentation strategies, network architecture, and optimizers [7]. We chose the ResNet18 as the backbone for experiments after comparing it with ResNet50 and WideResNet-28-2. All the model hyperparameters include an initial learning rate η of 0.03, SGD momentum β of 0.9, and weight decay set at 0.0005. Regarding the confidence threshold for pseudolabeling, most models employed τ = 0.95 and a labeled to unlabeled data ratio (μ) of 7. All the experiments were conducted using a single NVIDIA A5000 with 24GB of VRAM.

TABLE II
PERFORMANCE FOR DIFFERENT BACKBONES ON UCM AND AID

Model	Dataset	Top1	Best-Top1	GPU (GB)	Training Time (hours)
RN18	UCM	92.48	94.38	3.5	82:30
RN50	UCM	93.43	94.38	7	127:31
WRN	UCM	93.62	95.14	12	132:35
RN18	AID	91.04	91.32	3.5	261:09
RN50	AID	90.9	91.66	7	263:18
WRN	AID	93.08	94.66	12	266:19

Note: ”Best-Top 1” is the highest model performance during test set training and ”Top 1” is the accuracy after the model fully converges at epoch 512.

IV. RESULTS AND DISCUSSION

We first test the performance of CoMatch with different architectures on the original datasets and then evaluate CoMatch+CCSSL and CoMatch+Triplet on reduced datasets.

A. Backbone comparison for CoMatch algorithm.

In our comparative analysis, we evaluated three models —ResNet18 (RN18), ResNet50 (RN50), and WideResNet-28-2 (WRN)— utilizing the CoMatch algorithm on the original datasets. On the UCM dataset, RN18 and RN50 achieved a best-top 1 accuracy of 94.38%, while WRN outperformed with 95.14%. Shifting to the AID dataset, a minor accuracy decrease is observed, attributed to the dataset increased class complexity. Nevertheless, substantial accuracies persist, with RN18 at 91.32% and RN50 slightly higher at 91.66%. Notably, WRN achieved an outstanding 94.66% best-top1 (Table II).

RN18 demonstrated efficient GPU memory consumption, using only 3.5 GB and achieving the shortest training times. In contrast, RN50 consumed 7 GB with moderate training durations. WRN, known for high performance, utilized the highest GPU memory at 12 GB, requiring the longest training times. WideResNet-28-2 excelled in classification accuracy, but ResNet18 was accurate and resource-efficient, making it the suitable choice for further comparisons.

B. Performance of trainers on a reduced dataset.

We evaluated the performance of CoMatch, CoMatch+CCSSL, and CoMatch+Triplet in a constrained scenario using ResNet18 for 512 epochs. The results in Table III highlight the impact of different training methods on accuracy and training times. Notably, CoMatch+Triplet achieved the best-top 1 with 93.2% on UCM, while CoMatch and CoMatch+CCSSL showed slight reductions in accuracy. On the AID dataset, accuracy remained consistently high, staying close to 90%, across all trainers. These high accuracies indicate that these benchmark datasets represent a somewhat affordable challenge for all trainers, with limited room for substantial improvement. In terms of training computational time, most trainers demonstrated comparable extended durations, averaging around 200 hours for UCM and 250 hours for AID. Notably, the reduction in the parameter μ for CoMatch+CCSSL In-Distribution resulted in a significant decrease in training duration, ranging from 75 to 95 hours.

TABLE III
PERFORMANCE FOR DIFFERENT TRAINERS ON UCM AND AID

Trainer	Dataset	μ	Top1	Best-Top1	Training Time
CoMatch	UCM	7	90.4	92.8	205:59
CCSSL$_{ID}$	UCM	5	90.4	92	112:38
CCSSL$_{OD}$	UCM	7	89.4	92	205:39
Triplet	UCM	7	92.6	93.2	206:41
CoMatch	AID	7	91.49	92.19	247:58
CCSSL$_{ID}$	AID	5	90.73	91.05	173:17
CCSSL$_{OD}$	AID	7	89.46	90.22	248:33
Triplet	AID	7	86.16	90.92	248:35

Note: ID and OD correspond to In and Out of Distribution, respectively.

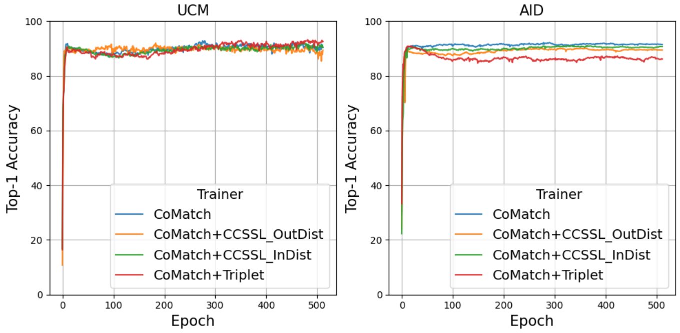

Fig. 4. Classification Accuracy per epoch on UCM and AID.

C. Confusion matrix and embeddings visualization for AID.

We aimed to evaluate the model’s classification accuracy for each class. The original AID dataset with 30 classes showed 8-15 instances of confusion among them. By retaining the 10 classes with more mistakes, we pushed the model’s classification limits. In Figure 5, we present the confusion matrix for AID, selected because it contains more samples per class. Notably, Airport-RailwayStation exhibits the highest confusion when using CM+Triplet. Interestingly, CM+Triplet managed to reduce confusion between Playground-Stadium, a challenging-to-distinguish class.

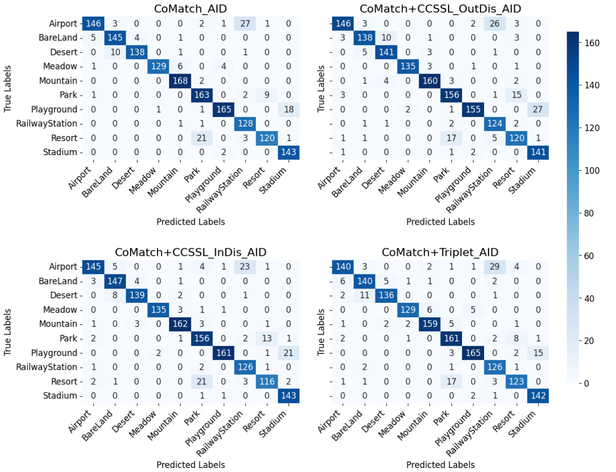

Fig. 5. Confusion matrices on the AID test dataset with 10 classes.

To visualize the embedding space, we applied t-Stochastic Neighborhood Embedding (t-SNE) to the 64-dimensional embeddings from the test dataset. We maintained this dimensionality across all models, as increasing it can potentially improve accuracy [7]. Notably, by the 10th epoch, all classes were already well-clustered, indicating limited improvement beyond this point as shown in figure 6. Note how the CoMatch-Triplet produces a clearer separation of the different class samples in the embedding space at very early stages, being able to learn faster and get maximum performance with few iterations, as it can be seen in figure 4.

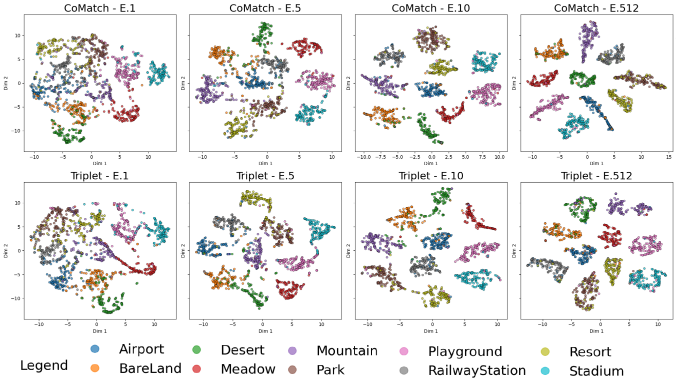

Fig. 6. tSNE plots (perplexity 30 and 300 iterations) for AID embeddings at epochs 1, 5, 10, and 512 of CoMatch and CoMatch+Triplet.

V. Conclusion

This study explores leveraging unlabeled data in remote sensing through advanced semi-supervised and self-supervised learning. We expanded the CoMatch algorithm by incorporating the existing Class-Aware Contrastive Semi-Supervised Learning (CCSSL) framework and introducing a deep metric learning triplet loss. In experiments conducted on reduced datasets with high-confusion classes, our findings highlight CoMatch's robustness and notable accuracies. CCSSL focuses on exploring dataset separation, both in and out of distribution, while the triplet loss's inclusion aims at embedding space separation and facilitating faster model convergence. This research contributes to remote sensing applications, emphasizing the effectiveness of advanced learning strategies in utilizing unlabeled data for image classification tasks.

VI. References Section

References

[1] I. Hernandez-Sequeira, R. Fernandez-Beltran, Y. Xu, P. Ghamisi, and F. Pla, "Semi-supervised classification for remote sensing datasets," in *Image Analysis and Processing – ICIAP 2023*, G. L. Foresti, A. Fusiello, and E. Hancock, Eds. Cham: Springer Nature Switzerland, 2023, pp. 463–474.

[2] Y. Yang and S. 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*, ser. GIS '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 270–279. [Online]. Available: https://doi.org/10.1145/1869790.1869829

[3] G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, and X. Lu, "AID: A Benchmark Data Set for Performance Evaluation of Aerial Scene Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3965–3981, Jul. 2017. [Online]. Available: https://ieeexplore.ieee.org/document/7907303

[4] J. Li, C. Xiong, and S. C. H. Hoi, "Comatch: Semi-supervised learning with contrastive graph regularization," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9455–9464. [Online]. Available: https://ieeexplore.ieee.org/document/9710505

[5] F. Yang, K. Wu, S. Zhang, G. Jiang, Y. Liu, F. Zheng, W. Zhang, C. Wang, and L. Zeng, "Class-aware contrastive semi-supervised learning," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 14 401–14 410. [Online]. Available: https://ieeexplore.ieee.org/document/9880146

[6] L. Schmarje, M. Santarossa, S.-M. Schroder, and R. Koch, "A Survey on Semi-, Self- and Unsupervised Learning for Image Classification," *IEEE Access*, vol. 9, pp. 82 146–82 168, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9442775/

[7] K. Musgrave, S. Belongie, and S.-N. Lim, "A metric learning reality check," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 681–699.

[8] N. Jean, S. Wang, A. Samar, G. Azzari, D. Lobell, and S. Ermon, "Tile2vec: Unsupervised representation learning for spatially distributed data," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 3967–3974, Jul. 2019. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/4288

[9] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, "S4L: Self-Supervised Semi-Supervised Learning," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019, pp. 1476–1485. [Online]. Available: https://ieeexplore.ieee.org/document/9010283

[10] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, "FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 596–608. [Online]. Available: https://papers.nips.cc/paper/2020/hash/06964dce9addb1c5cb5d6e3d9838f733-Abstract.html

[11] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality Reduction by Learning an Invariant Mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, Jun. 2006, pp. 1735–1742. [Online]. Available: https://ieeexplore.ieee.org/document/1640964

[12] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum Contrast for Unsupervised Visual Representation Learning," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, Jun. 2020, pp. 9726–9735. [Online]. Available: https://ieeexplore.ieee.org/document/9157636/

[13] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation Learning with Contrastive Predictive Coding," Jan. 2019. [Online]. Available: http://arxiv.org/abs/1807.03748

[14] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 1597–1607. [Online]. Available: https://proceedings.mlr.press/v119/chen20j.html

[15] K. Q. Weinberger, J. Blitzer, and L. Saul, "Distance Metric Learning for Large Margin Nearest Neighbor Classification," in *Advances in Neural Information Processing Systems*, vol. 18. MIT Press, 2005. [Online]. Available: https://proceedings.neurips.cc/paper/2005/hash/a7f592cef8b130a6967a90617db5681b-Abstract.html

[16] G. Sumbul, M. Ravanbakhsh, and B. Demir, "Informative and Representative Triplet Selection for Multilabel Remote Sensing Image Retrieval," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–11, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9594829

[17] H. Jung, Y. Oh, S. Jeong, C. Lee, and T. Jeon, "Contrastive Self-Supervised Learning With Smoothed Representation for Remote Sensing," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9397864/

[18] J. Kang, R. Fernandez-Beltran, P. Duan, S. Liu, and A. J. Plaza, "Deep Unsupervised Embedding for Remotely Sensed Images Based on Spatially Augmented Momentum Contrast," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 3, pp. 2598–2610, Mar. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9140372

[19] K. Musgrave, S. J. Belongie, and S.-N. Lim, "Pytorch metric learning," *ArXiv*, vol. abs/2008.09164, 2020.