Simple and Robust Multi-Label Object Detection in Remote-Sensing Images via Self-Supervised Contrastive Learning

Luke Briody
Department of Computer Science
Brown University
lbriody@cs.brown.edu

Gun Kaan Aygen
Department of Computer Science
Brown University
gaygen@cs.brown.edu

Abstract

We evaluate self-supervised contrastive learning for use in multi-label object detection, particularly in remote-sensing images. Prior works in this area have claimed that contrastive learning methods originally designed for single-label settings cannot be trivially adapted to enhance a model's robustness in multi-label settings. We test this claim empirically by performing an ablation study comparing object detection capabilities of linear models that take encodings from either a frozen baseline encoder or the same baseline encoder pretrained with the SimCLR paradigm for self-supervised contrastive learning [3]. We show that a simple adaptation of self-supervised contrastive learning to the multi-label setting, in contrast with previous claims, greatly enhances object detection as measured by mean average precision; the contrastive pretraining paradigm increased mAP to 0.9133 from a baseline mAP of 0.6361 when tested on a diverse set of remote-sensing images. Our result highlights the necessity for a stronger understanding of contrastive learning, especially regarding theoretical guarantees of contrastive learning and how they might differ when applied in single- versus multi-label settings.

1 Introduction

While learning effective visual representations in noisy settings has been effectively addressed through supervised approaches, the growing volume of imaging datasets and expense of manual dataset annotation make *self*-supervised approaches a more attractive option. Most mainstream, self-supervised approaches to learning robust visual representations may be categorized as either generative or discriminative. Generative approaches, which directly synthesize or model pixels in the input space, have shown promise in learning robust representations. That being said, pixel-level synthesis is a great computational expense and seems not to be strictly necessary for representation learning. Indeed, self-supervised discriminative approaches provide a strong alternative, learning robust visual representations with less computational overhead. Typical discriminative approaches

1. train models to perform handcrafted pretext tasks (e.g. solving jigsaw puzzles or predicting rotations/augmentations) [5; 8; 14; 24], or
2. remodel the local structure of a model's latent space such that the distance between embeddings decreases for positive pairs and increases for negative pairs [2; 6; 9; 20].

The second approach is known as contrastive learning (CL). Due to the largely heuristic nature of handcrafted pretext tasks and the state-of-the-art performance of many approaches based on contrastive learning, we focus our efforts in this work on self-supervised contrastive learning.

Preprint. Under review.

Despite the success of self-supervised contrastive learning-based approaches when applied to single-label object detection and classification tasks, to our knowledge, such approaches have not been applied to tasks where a single image may have multiple positive labels. Rather, in these multi-label settings, techniques using supervised versions of contrastive learning appear to dominate the literature [1; 4; 22; 23]. These multi-label techniques are typically more involved than the direct reuse of an approach originally tested on single-label tasks. What is more, some authors go as far as explicitly declaring that the trivial adaptation of a single-label approach would not be a tenable approach to improve the quality of learned visual representations for multi-label settings; exemplifying this, Dao et al. [4] claim:

> “Given the appealing properties and promising results of CL in single-label classification, it is natural to adapt it into multi-label cases to boost performance. However, this adaptation is non-trivial.”

Dao et al. [4] proceed by laying out a heuristic argument for why this ought to be the case but offer neither empirical data nor rigorous theoretical proof to support their line of reasoning. Given the obvious advantages that would belie a self-supervised contrastive learning approach capable of operating in both single- and multi-label settings, our aims in this work are twofold. Particularly, we seek to study whether:

1. Self-supervised contrastive learning enables robust visual representation learning in a multi-label setting.
2. The trivial adaptation of a contrastive learning approach from a single-label setting to a multi-label setting can learn improved visual representations.

To address these aims, we trivially adapted the SimCLR paradigm [3], a well-known approach to self-supervised contrastive learning originally designed for and tested on single-label tasks, for downstream use in multi-label object detection in remote-sensing images. We performed an ablation study to determine if the SimCLR paradigm improved the quality of learned visual representations, as measured by performance on the object detection task. The exact details of the evaluation protocol are described in Section 2.4.

We specifically evaluate object detection performance on a dataset consisting of remote-sensing images because this field’s needs dovetail nicely with the aims of our research aims. In particular, remote-sensing image datasets are becoming increasingly large (and hence, increasingly expensive to annotate), so this area stands to benefit greatly from applications of self-supervised learning. Moreover, remote-sensing image datasets are inherently noisy for a variety of reasons [19], so learning robust visual representations is paramount. Specifically, some potential temporal changes to the environment that cause satellite imagery to be inherently noisy are temporally varying lighting (e.g. due to time of day) and seasonal conditions, atmospheric phenomena, and land cover (e.g. snow and/or ice). In addition, the varying spatial resolution of imaging sensors and the degree of noise different imaging sensors capture are also significant reasons for the need for a noise-invariant approach in multi-label object detection in remote sensing.

2 Ablation Study

The protocol for our ablation study comprises three key steps:

1. pretraining an experimental encoder using the SimCLR paradigm,
2. training object detection models (one whose inputs are the output of a baseline encoder and one whose inputs are the output of the experimental encoder), and
3. evaluating the best iteration of each object detection model on an unseen test dataset.

The baseline encoder (Section 2.2) serves as a negative control in our analyses, and its architecture and parameters are identical to those of the experimental encoder (Section 2.3) before the self-supervised contrastive pretraining pbase of our protocol. All experiments were scripted in the Python programming language using the PyTorch framework for deep learning [15] and run on a single NVIDIA L4 Tensor Core graphics processing unit. Our code is available on GitHub.[1]

[1] `https://github.com/lbriody/simple-multilabel-cl`

2.1 Dataset

We use the MLRSNet [16] dataset for training, validating, and evaluating the baseline and experimental models. MLRSNet consists of 109,161 remote-sensing images divided into 46 scene categories. Each image in MLRSNet is further annotated to indicate the presence of instances of the 60 predefined object classes. Per image, there are at least one and as many as 13 positive labels among the object classes. Example images for most scene categories and object classes are shown below in Figure 1. Critically, the images in MLRSNet are of high resolution–image resolution varies in the range of 0.01 to 100 square meters per pixel–and the dataset was empirically determined to be more diverse than two preexisting benchmark datasets for remote sensing in terms of viewpoint, object pose, illumination, background, and presence of occlusions [16].

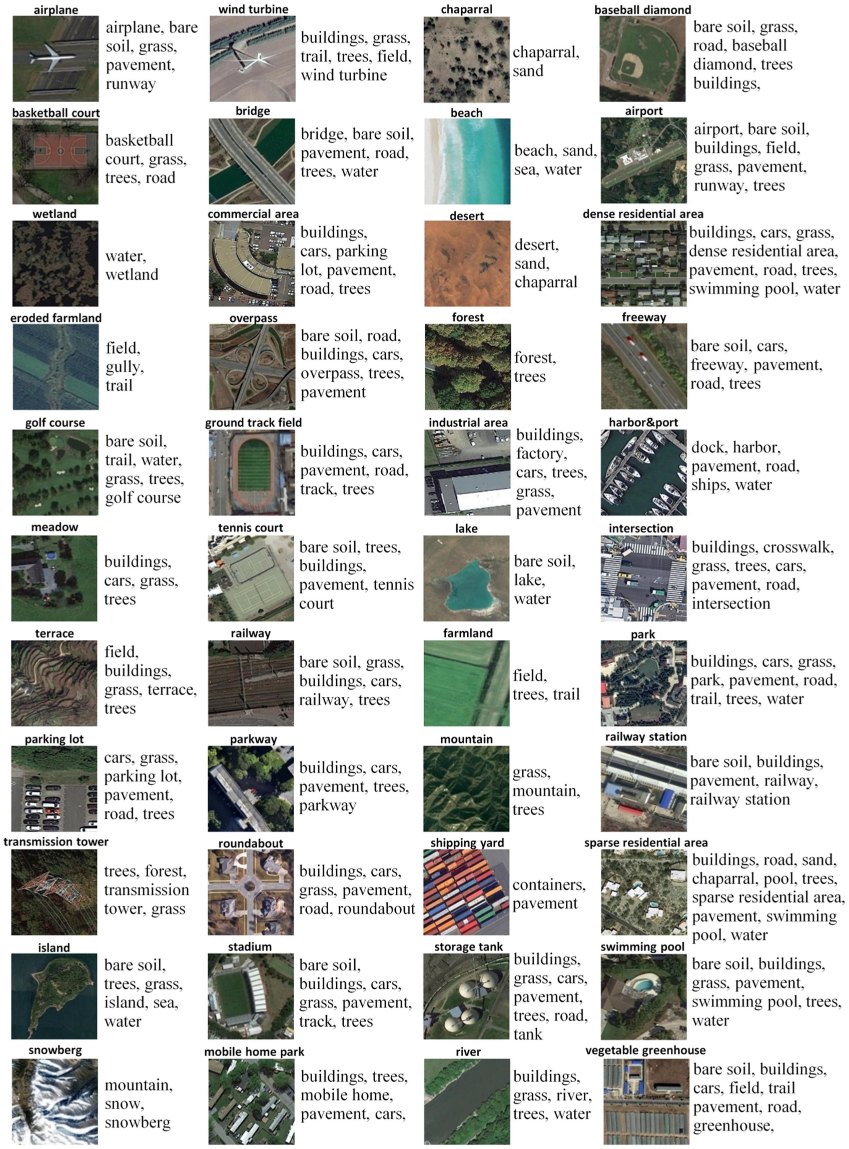
airplane
airplane, bare soil, grass, pavement, runway
wind turbine
buildings, grass, trail, trees, field, wind turbine
chaparral
chaparral, sand
baseball diamond
bare soil, grass, road, baseball diamond, trees buildings,
basketball court
basketball court, grass, trees, road
bridge
bridge, bare soil, pavement, road, trees, water
beach
beach, sand, sea, water
airport
airport, bare soil, buildings, field, grass, pavement, runway, trees
wetland
water, wetland
commercial area
buildings, cars, parking lot, pavement, road, trees
desert
desert, sand, chaparral
dense residential area
buildings, cars, grass, dense residential area, pavement, road, trees, swimming pool, water
eroded farmland
field, gully, trail
overpass
bare soil, road, buildings, cars, overpass, trees, pavement
forest
forest, trees
freeway
bare soil, cars, freeway, pavement, road, trees
golf course
bare soil, trail, water, grass, trees, golf course
ground track field
buildings, cars, pavement, road, track, trees
industrial area
buildings, factory, cars, trees, grass, pavement
harbor&port
dock, harbor, pavement, road, ships, water
meadow
buildings, cars, grass, trees
tennis court
bare soil, trees, buildings, pavement, tennis court
lake
bare soil, lake, water
intersection
buildings, crosswalk, grass, trees, cars, pavement, road, intersection
terrace
field, buildings, grass, terrace, trees
railway
bare soil, grass, buildings, cars, railway, trees
farmland
field, trees, trail
park
buildings, cars, grass, park, pavement, road, trail, trees, water
parking lot
cars, grass, parking lot, pavement, road, trees
parkway
buildings, cars, pavement, trees, parkway
mountain
grass, mountain, trees
railway station
bare soil, buildings, pavement, railway, railway station
transmission tower
trees, forest, transmission tower, grass
roundabout
buildings, cars, grass, pavement, road, roundabout
shipping yard
containers, pavement
sparse residential area
buildings, road, sand, chaparral, pool, trees, sparse residential area, pavement, swimming pool, water
island
bare soil, trees, grass, island, sea, water
stadium
bare soil, buildings, cars, grass, pavement, track, trees
storage tank
buildings, grass, cars, pavement, trees, road, tank
swimming pool
bare soil, buildings, grass, pavement, swimming pool, trees, water
snowberg
mountain, snow, snowberg
mobile home park
buildings, trees, mobile home, pavement, cars,
river
buildings, grass, river, trees, water
vegetable greenhouse
bare soil, buildings, cars, field, trail pavement, road, greenhouse,

Figure 1: Exemplary images and corresponding positive object-class labels for 44 of the 46 categories found in the MLRSNet dataset (excluding only 'bareland' and 'cloud'). The label above each image records the appropriate scene category, and the labels to the right of each image record the object classes pictured. [16]

We performed (pseudo-)random sampling to partition the full dataset into (pre)training, validation, and test datasets consisting of 70, 20, and 10 percent of the full dataset, respectively.

2.2 Baseline Encoder

For our baseline model, we adopted the widely used ResNet-50 [10], pretrained on ImageNet ILSVRC-2012 [17]. Particularly, we accessed the best available pretrained model parameters for ResNet-50 via the Torchvision Multi-Weight API (`IMAGENET1K_V2`). Excluding DenseNet models [11] due to the large memory overhead of the necessary concatenation operations, ResNet-50 generally performed better than other architectures tested on MLRSNet in terms of both mean average precision and F_1 scores [16], and ResNet-50 was utilized as the baseline encoder by Chen et al. [3] in their original exposition of the SimCLR paradigm, making it optimal for our purposes. Since we need only the visual representations learned by ResNet-50, we remove the classification head from the model, instead taking the output after the average pooling layer as an input image's encoding.

2.3 Experimental (SimCLR) Encoder

For our experimental encoder, we take an identical copy of the baseline encoder and apply the SimCLR pretraining paradigm (Figure 2). Briefly, for a batch of size k, the SimCLR paradigm proceeds as follows: for each image $\mathbf{x}$ in the batch, two augmented views of the image $\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j$ are created by applying transformations t, t' belonging to some family of augmentations $\mathcal{T}$. A complete description of $\mathcal{T}$ is given in Section 2.3.1. Then passed through the encoder, f. The two output embeddings, $\mathbf{h}_i = f(\tilde{\mathbf{x}}_i)$ and $\mathbf{h}_j = f(\tilde{\mathbf{x}}_j)$, are flattened and passed through a multi-layer perceptron (MLP), g. The structure of the MLP g is the same as in Chen et al. [3]. Letting $\mathbf{z}_i = g(\mathbf{h}_i)$ and $\mathbf{z}_j = g(\mathbf{h}_j)$, the contrastive loss attributed to the positive pair (i, j) is

$$\ell_{i,j} = -\log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau))}{\sum_{t=1}^{2k} \mathbb{1}_{[t \neq i]} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_t)/\tau)};$$

wherein

- $\text{sim}(\cdot, \cdot)$ is the cosine similarity function,
- $\mathbb{1}_{[t \neq 1]} \in \{0, 1\}$ is an indicator function evaluating to 1 if and only if $t \neq i$, and
- τ is a temperature parameter.

The contrastive loss is finally backpropagated through g and f. From the formulation of $\ell_{i,j}$, the essential insight of the SimCLR paradigm becomes apparent. Particularly, if k is large enough, then fixing a positive pair (i.e. augmented copies of the same image), each combination of an image in the positive pair and one of the remaining $2(k-1)$ images in the batch can be used as 'implicit' negative pairs. This does away with the need for explicit negative hard mining [3].

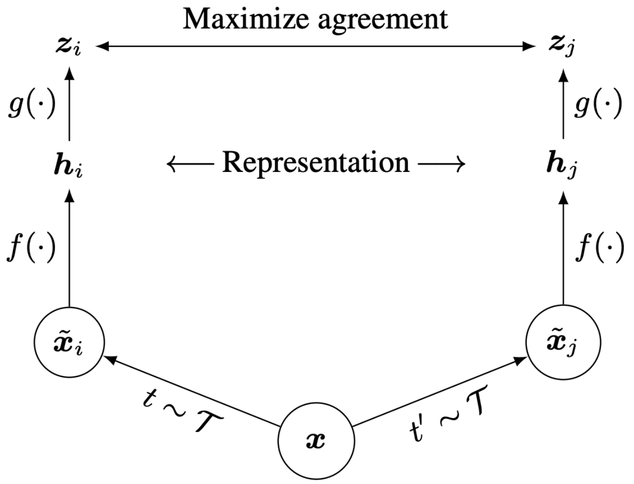

Figure 2: Schematic representation of the SimCLR pretraining paradigm [3].

2.3.1 Augmentations

Each augmentation in the family of augmentations $\mathcal{T}$ (from which we randomly sample two sets of transformations t, t' and apply to an image $\mathbf{x}$ to obtain two views $\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j$) consists of a sequence of the following form:

- cropping to a random size, then rescaling to the appropriate input size,
- flipping horizontally with a probability of 0.5,
- color jittering with a probability of 0.8,
- converting to grayscale with a probability of 0.2,
- applying a Gaussian blur with a 23×23 kernel and standard deviation randomly sampled from the interval $[0.1, 2.0]$, and
- normalization using the training dataset's mean and standard deviation.

Color jittering comprises, more specifically, adjusting four photometric quantities (brightness, contrast, saturation, and hue) by a random value selected uniformly at random from an interval specified for each quantity. For brightness, contrast, and saturation, the selection interval is specified as $[0.6, 1.4]$; for hue, the selection interval is specified as $[-0.1, 0.1]$. These augmentations and parameters were chosen based on the empirical tests performed in the original SimCLR publication to determine the optimal transformation profile. Toy examples of these augmentations are shown in Figure 3.

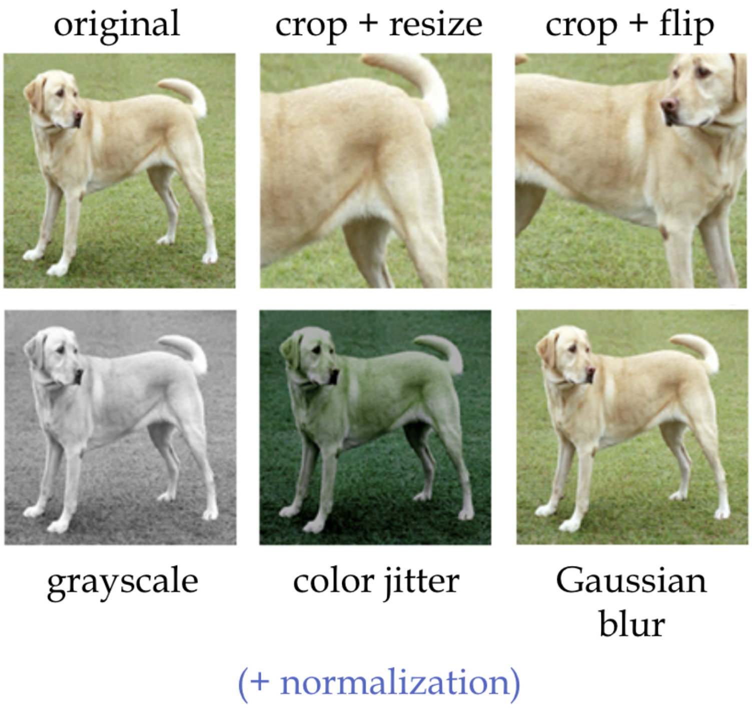

Figure 3: Toy examples of augmentations used in the SimCLR pretraining paradigm [3].

2.3.2 Pretraining Configuration

The self-supervised pretraining phase for the experimental encoder was given a duration 10 epochs with a batch size of 128 (further discussed in Section 4.1). The temperature parameter was set at $\tau = 0.5$, and the contrastive loss was optimized with the Adam optimizer and a learning rate of 3E-4.

2.4 Linear Evaluation Protocol

We follow a modified version of the widely used linear evaluation protocol [2; 12; 20; 24] to evaluate the quality of the encoders' learned visual representations in the multi-label setting. In other terms, to compare the baseline and experimental encoders, a 60-node linear probe was trained atop each encoder to detect the presence of each of the 60 pre-defined object classes of MLRSNet. The linear probes were trained for 10 epochs with a batch size of 128. Binary cross-entropy loss was computed and backpropagated through the linear probes only; encoder weights were frozen for the duration of this phase of the protocol. The linear probe's parameters were optimized with the Adam optimizer using a learning rate of 1E-3 to minimize binary cross-entropy loss.

2.4.1 Evaluation Metrics

Mean average precision (mAP) was used as the primary metric to compare the object-detection performance of the baseline model against the experimental model. Mean average precision is widely used as an evaluation metric for multi-label object-detection tasks; by and large, it is the community standard for quantifying performance in such a setting. Indeed, mAP is implemented as the standard evaluation metric for well-reputed object-detection challenges such as Microsoft COCO [13] and Pascal VOC [7]. Briefly, mAP is computed as follows. Provided with N images, for the kth object class k, sort the predicted N-dimensional vector of logits output by a model in decreasing order. The average precision for the kth class (AP$_k$) is then computed as

$$\mathrm{AP}_k = \sum_{n=1}^{N} P_{k,n} \cdot (R_{k,n} - R_{k,n-1})\,;$$

where

- $R_{k,n}$ is the recall score for the kth class and n^{th} image,
- $R_{k,n-1}$ is the recall score for the kth class and $(n-1)^{\text{th}}$ image, and
- $P_{k,n}$ is the precision score for the kth class and n^{th} image.

Finally, the mean of average precision scores over all 60 pre-defined object classes is computed, yielding mAP:

$$\mathrm{mAP} = \frac{1}{k} \sum_{k=1}^{60} \mathrm{AP}_k.$$

Though mAP is the primary evaluation metric in this study, it is non-differentiable [21], so we take the common approach of minimizing binary cross-entropy loss as a proxy for maximizing mAP.

3 Results

The mAP metric values evaluated on the test split demonstrate that the trivial adaptation of the SimCLR paradigm for the experimental model outperforms the baseline model by a considerable margin. After ten epochs of training the baseline model's downstream linear classifier with binary cross-entropy loss, the baseline model achieves 0.6361 test mAP. For the experimental model, after 10 epochs of SimCLR pretraining and 10 epochs of training for the linear classifier with binary cross-entropy loss and the encoder frozen, 0.9133 mAP is achieved. Furthermore, the binary cross-entropy loss for the experimental model is much lower than for the baseline model throughout all ten epochs of training and validation (Figure 4, overleaf), and the experimental model's mAP is considerably lower than that of the baseline model through all ten epochs of validation (Figure 5, overleaf).

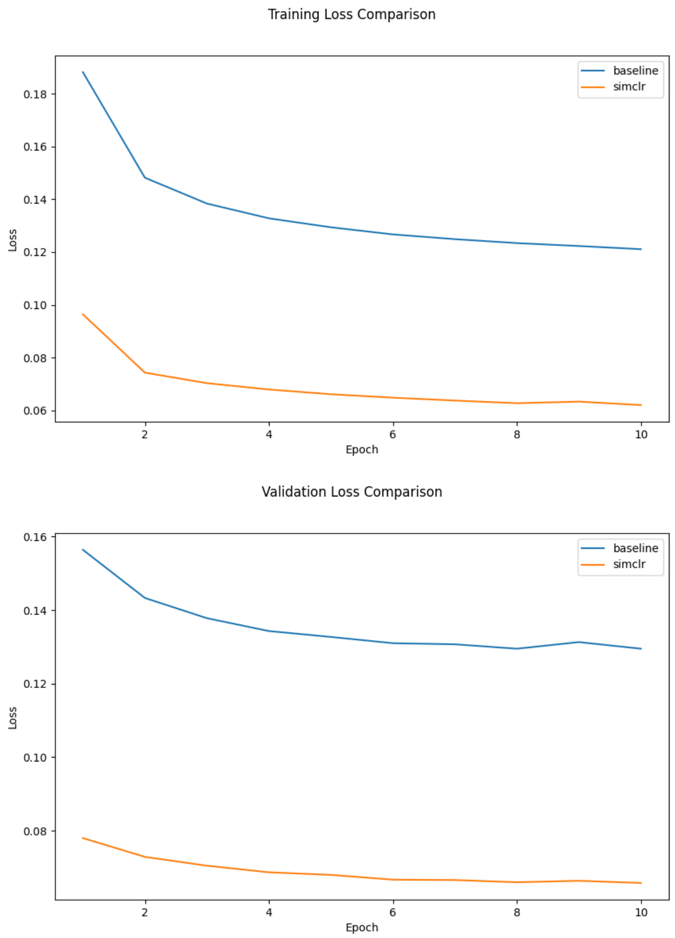

Figure 4: Comparison of binary cross-entropy losses between baseline and experimental models during linear evaluation. Top: training losses. Bottom: validation losses.

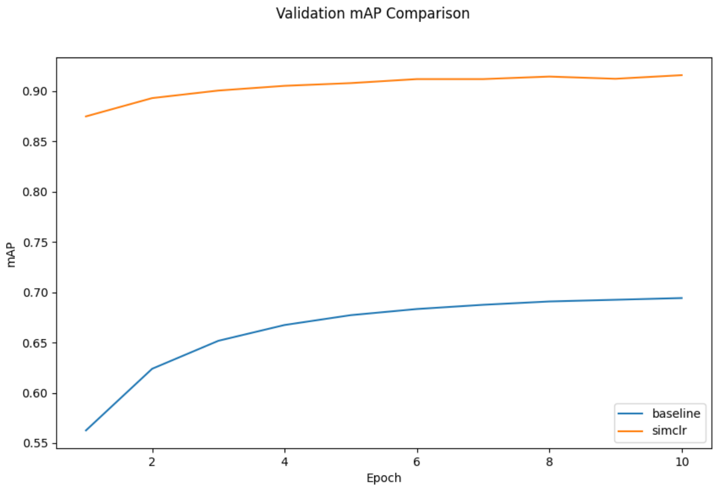

Figure 5: Comparison of mAP between baseline and experimental models during linear evaluation.

4 Discussion

4.1 Limitations and Considerations

At face value, our experiments may appear to be limited due to the sample complexity of the MLRSNet dataset. The number of images in the dataset is approximately 110,000. Given the complexity of the task of semantic understanding in multi-object scenes, training on a larger remote-sensing dataset such as BigEarthNet [18] (which contains approximately 590,000 remote-sensing patches) may be better suited to produce an industry-standard model for object detection in remote-sensing images. However, BigEarthNet falls short of MLRSNet in terms of several key metrics. For one, BigEarthNet lacks a diversity of spatial resolution; all images in the dataset were taken at a resolution of 100 square meters per pixel. Moreover, BigEarthNet comprises only 19 classes compared to MLRSNet's 60 classes. Hence, while MLRSNet might not be the optimal training dataset for real-world applications of our work, because MLRSNet consists of fewer images with greater variation in camera geometry and image contents, utilizing this dataset in our empirical study better demonstrates the robustness of the experimental model.

Another limitation we faced stemmed from the available computational resources for training. We trained our models for ten epochs on Google Colab using an NVIDIA L4 GPU. The maximum batch size we could use in training our models was 128, where 22.5/22.7 GB of available GPU RAM was utilized in training. Any attempt at increasing the batch size beyond this limit led to memory-related exceptions at runtime. Because Chen et al. [3]'s key insight in introducing the SimCLR paradigm was that large batch sizes can allow one to circumvent the difficult problem of hard negative mining, it would have been ideal to use a larger batch size (e.g. 2048) during the contrastive pretraining phase in particular. Because of this strictly suboptimal pretraining routine, we interpret the performance of our experimental model as a lower bound on the performance of a model pretrained to convergence with a larger batch size. Nevertheless, the experimental model significantly and conclusively outperformed the baseline model in the multi-label object detection task, so this limitation was not entirely detrimental to our project.

4.2 Future Work

Future iterations of this research should adapt the self-supervised contrastive pretraining paradigm non-trivially to the multi-label object detection task, and compare whether the non-trivial extension outperforms the trivial extension implemented in our paper, as we did not have the computational resources available to make such a comparison accurately. It would be particularly intriguing to see what differences, if any, appear in local and global latent information structures of image encodings depending on whether a trivial or non-trivial adaptation of a contrastive learning paradigm is used. For exploring local and global information structures, respectively, the application of techniques such as t-distributed stochastic neighborhood embedding (t-SNE) and uniform manifold approximation and projection (UMAP) could provide relevant insights.

5 Conclusion

All in all, our project demonstrates that the foundational understanding of approaches to contrastive learning–particularly as they apply in multi-label settings–requires further scrutiny. Though it remains to empirically compare trivial and non-trivial adaptations of contrastive learning in multi-label settings, our ablation study indicates that the heuristic arguments in opposition to trivial adaptations may be too quick to judge, especially given the key advantage that while non-trivial adaptations require supervision, trivial adaptations may be self-supervised. A unified theoretical analysis of self-supervised and supervised approaches to contrastive learning in multi-label settings, trivial and non-trivial, is warranted to empower the research community and industry alike with accurate information regarding the exact benefits and drawbacks of these various strategies.

References

[1] A. Audibert, A. Gauffre, and M.-R. Amini. Multi-label contrastive learning : A comprehensive study, 2024. URL `https://arxiv.org/abs/2412.00101`.

[2] P. Bachman, R. D. Hjelm, and W. Buchwalter. Learning representations by maximizing mutual information across views, 2019. URL `https://arxiv.org/abs/1906.00910`.

[3] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations, 2020. URL `https://arxiv.org/abs/2002.05709`.

[4] S. D. Dao, E. Zhao, D. Phung, and J. Cai. Multi-label image classification with contrastive learning, 2021. URL `https://arxiv.org/abs/2107.11626`.

[5] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction, 2016. URL `https://arxiv.org/abs/1505.05192`.

[6] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks, 2015. URL `https://arxiv.org/abs/1406.6909`.

[7] M. Everingham, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.*, 88(2):303–338, 2010. URL `http://dblp.uni-trier.de/db/journals/ijcv/ijcv88.html#EveringhamGWWZ10`.

[8] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations, 2018. URL `https://arxiv.org/abs/1803.07728`.

[9] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2:1735–1742, 2006. URL `https://api.semanticscholar.org/CorpusID:8281592`.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015. URL `https://arxiv.org/abs/1512.03385`.

[11] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks, 2018. URL `https://arxiv.org/abs/1608.06993`.

[12] A. Kolesnikov, X. Zhai, and L. Beyer. Revisiting self-supervised visual representation learning, 2019. URL `https://arxiv.org/abs/1901.09005`.

[13] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár. Microsoft coco: Common objects in context, 2015. URL `https://arxiv.org/abs/1405.0312`.

[14] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles, 2017. URL `https://arxiv.org/abs/1603.09246`.

[15] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019. URL `https://arxiv.org/abs/1912.01703`.

[16] X. Qi, P. Zhu, Y. Wang, L. Zhang, J. Peng, M. Wu, J. Chen, X. Zhao, N. Zang, and P. T. Mathiopoulos. Mlrsnet: A multi-label high spatial resolution remote sensing dataset for semantic scene understanding, 2020. URL `https://arxiv.org/abs/2010.00243`.

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

[18] G. Sumbul, A. de Wall, T. Kreuziger, F. Marcelino, H. Costa, P. Benevides, M. Caetano, B. Demir, and V. Markl. Bigearthnet-mm: A large-scale, multimodal, multilabel benchmark archive for remote sensing image classification and retrieval [software and data sets]. *IEEE Geoscience and Remote Sensing Magazine*, 9(3):174–180, Sept. 2021. ISSN 2373-7468. doi: 10.1109/mgrs.2021.3089174. URL `http://dx.doi.org/10.1109/MGRS.2021.3089174`.

[19] X. Tang, R. Du, J. Ma, and X. Zhang. Noisy remote sensing scene classification via progressive learning based on multiscale information exploration. *Remote Sensing*, 15(24), 2023. ISSN 2072-4292. doi: 10.3390/rs15245706. URL `https://www.mdpi.com/2072-4292/15/24/5706`.

[20] A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding, 2019. URL `https://arxiv.org/abs/1807.03748`.

[21] B. Wang. A parallel implementation of computing mean average precision, 2022. URL `https://arxiv.org/abs/2206.09504`.

[22] V. Zaigrajew and M. Ziba. Contrastive learning for multi-label classification. 2022. URL `https://api.semanticscholar.org/CorpusID:259143904`.

[23] P. Zhang and M. Wu. Multi-label supervised contrastive learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(15):16786–16793, Mar. 2024. doi: 10.1609/aaai.v38i15.29619. URL `https://ojs.aaai.org/index.php/AAAI/article/view/29619`.

[24] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization, 2016. URL `https://arxiv.org/abs/1603.08511`.