RarePlanes Soar Higher: Self-Supervised Pretraining for Resource Constrained and Synthetic Datasets

Justin Downes, Will Gleave, Dan Nakada
Amazon Web Services
{jusdow,sgleave,nakadadn}@amazon.com

Abstract

Self-supervised pretraining has advanced the capabilities of many computer vision tasks without requiring additional labels. One drawback is this technique requires extensive datasets and computational resources. This requirement of large datasets to pretrain with has often precluded the use of smaller, more niche datasets. Recently a method of pretraining has been developed that uses several stages of training, arranging each subsequent pretraining step to a dataset more closely resembling the target labelled data. This Hierarchical PreTraining (HPT) allows small datasets that are significantly different from generalized pretraining datasets (e.g. ImageNet) to build off subsequent knowledge transfers of increasingly focused training. However, there remains computer vision domains that are sufficiently difficult to acquire data that the use of synthetic data to augment their training has become a common convention. This paper examines how Remote Sensing Imagery (RSI) datasets, both augmented with synthetic data and without, still benefit from HPT despite being a niche domain. We show the fine balance that must be maintained when pretraining with these small datasets through a series of experiments focused on isolating various training parameters. We also demonstrate how these techniques lead to model improvements over existing baselines with and without synthetic data. Given that HPT provides a straightforward process to increase performance, and synthetic data is a growing resource for dataset augmentation, these combined methods can enhance a wide variety of current and future computer vision tasks.

1. Introduction

Transfer learning is a powerful tool to transmit knowledge learned from one domain to another. It is useful to leverage the knowledge of domains that have abundant datasets for training in domains that have little, and thereby reduce the amount of computation needed [15]. Recently, self-supervised learning methods have begun to surpass supervised learning as the most effective way to pretrain models [18]. Due to the costs in labeling large datasets, transitioning from supervised to self-supervised pretraining has enabled advances in accuracy, with no extra labels, as well as the enablement of targeted domain adaptation for label scarce specialist domains [1].

Modern self-supervised pretraining techniques are often framed in the context of utilizing large datasets [8] and large batch sizes [4]. These constraints become problematic when working with datasets that have limited labels and not enough data to create batches of sufficient size. Traditionally, supervised pretraining on large generalized datasets is coupled with fine tuning on the target domain data, thereby offering a layered narrowing of training. This transfer learning from a generalist dataset to a narrow domain increased the efficiency of the labels in that target domain [14]. Using this narrowing of focus in training has also been shown to aid in self-supervised training [18] through Hierarchical Pretraining (HPT).

Remote sensing is a highly specialized domain [20] where collecting data and labels requires specialized skills and publicly available datasets have limited availability to conduct traditional pretraining. Some areas of remote sensing imagery and associated labelled data are so sparse that creating 3D synthetic representations of the real data has become an active area of research [21, 12, 11] and a viable alternative to labelling data. This synthetic data generation has high implementation costs due to a need for specialized 3D software and expert modeling knowledge. Even with these high barriers of generating synthetic data this method is a common component of the remote sensing community and should be of interest to any pretraining task. Remote sensing imagery lends itself to doing HPT in that these overhead images provide a narrowing down of focus on objects of interest intrinsically due to the relatively small size of the objects compared to the overall image. Crops of these large remote sensing images can be tailored to focus on or away from targets of interest more easily than typical general pretraining datasets which tend to have objects that encompass

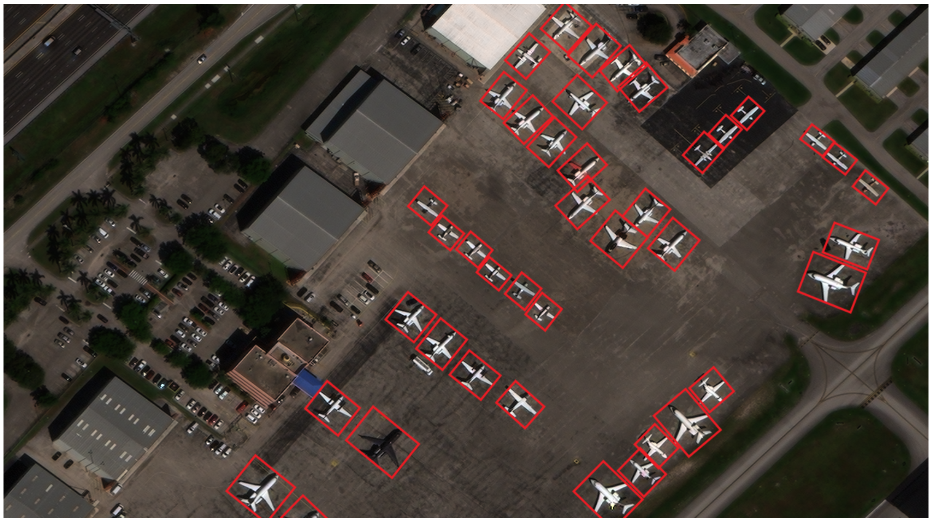


Figure 1: Example of the real RarePlanes data. Synthetic RarePlanes data shown in Figure 2

a larger portion of the image.

Our work covers the application of self-supervised pretraining techniques as they are applied to narrow RSI datasets and how the introduction of synthetic data augments this type of training. Specifically, we look at the HPT framework, as described in [18], implemented with the RarePlanes real and synthetic datasets [21] as well as the High Resolution SAR Images Dataset (HRSID) [24], and how the configurations generally accepted for large scale contrastive self-supervised learning [4] need to be adapted for much smaller datasets. Our main contributions are as follows:

- We find that **self-supervised** generalized pretraining outperforms **supervised** generalized pretraining. We also see increasing performance improvements as we layer on subsequent, more target aligned pretraining steps for small datasets.

- We show the narrow range in which HPT configurations can show improved performance over generalist self-supervised pretraining and how these models can quickly overfit for worse performance. Specifically, how smaller batch sizes and fewer training iterations perform better on these smaller datasets, which is counter to current self-supervised results with very large datasets [4].

- To the best of our knowledge, we're the first to study the incorporation of synthetic data in self-supervised pretraining and it's impact on downstream models. We demonstrate **self-supervised** pretraining with synthetic data improves model performance over traditional **supervised** pretraining but does not increase performance when pretraining on a mixture of real and synthetic data.

The paper is organized as follows. Section 2 covers related work in the self-supervised learning, synthetic overhead imagery, and the intersection of self-supervised learning techniques with synthetic data. Section 3 provides details of: the dataset, models used, and configuration of the training setup for our experiments. The results contained in section 4 provides an analysis of the different training parameters evaluated in isolation as well as the results of our synthetic experiments.

2. Related Work

Self-supervised learning is rapidly becoming the default method in pretraining models for transfer learning tasks. For visual features, these methods have been shown to offer improvements on generalized datasets [8], in narrow domains [1], and in transitioning pretraining hierarchically between the two [18]. Contrastive learning has been utilized to achieve state-of-the-art results in self-supervised pretraining [2, 23] and the introduction of a Simple Framework for Contrastive Learning of Visual Representations (SimCLR) [4] provides an even simpler methodology while achieving better results.

A common method to increase the amount of labeled data has been to integrate synthetically generated data into the training set. These synthetic generation mechanisms generally incorporate multiple stages, from initial 3D simulation, through various methods of domain adaptation [13, 22, 12], adding possible environmental simulations, and to a final rendered realistic scene. In the realm of overhead imagery these synthetic generation pipelines of-

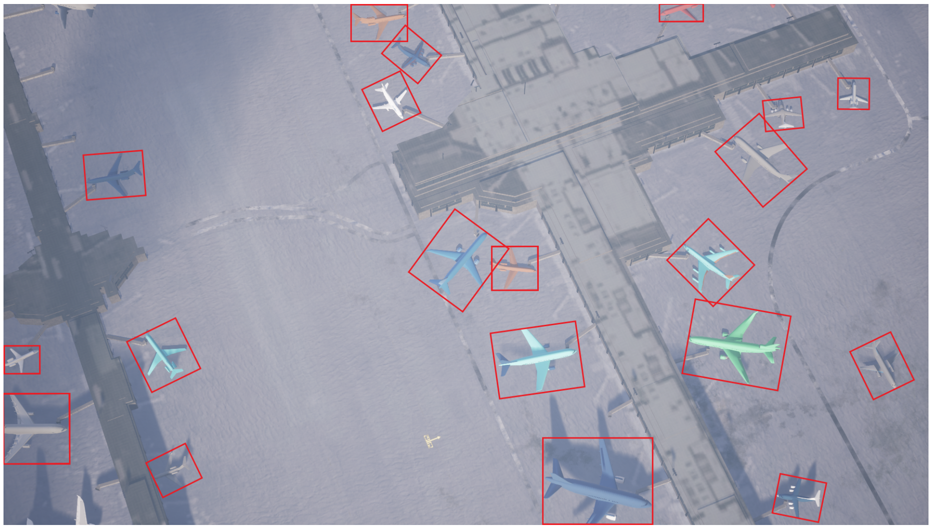


Figure 2: Example of the synthetic RarePlanes data. Real RarePlanes data shown in Figure 1

ten incorporate simulating the sensor platform [16] as well as integrating real locations [21, 10]. This modular pipeline is useful for separating out different techniques but relies heavily on domain expertise to put the right models together. We are currently seeing more automated approaches of scene generation especially with domain adaptation of entire scenes based on a stochastic element. This not only makes data more robust but also increases the amount of data that can be generated [5]. An end-to-end automated synthetic overhead scene generator is still out of reach, but is getting closer with the introduction of more powerful image generation and editing models [17].

While self-supervised learning seeks to capture feature representation through self-labeling methods [18, 7] and synthetic data can be thought of as hand crafting features for data augmentations [13, 5] in supervised tasks, the intersection of these two categories has not been thoroughly studied, even though they both can be used to improve downstream supervised tasks. Some of the current work leveraging both of these techniques include; using self-supervised learning on wholly synthetic datasets with synthetic evaluation [9] and as a comparison self-supervised training on real data to supervised training on synthetic data [6]. Of specific interest to our task is understanding if these hand crafted synthetic features aid the HPT task of narrowing down targeted self-supervised pretraining.

3. Method

We conduct a number of experiments to better understand self-supervised pretraining on a limited remote sensing dataset. We follow the general workflow of Reed *et al*. [18], using generalist, specialist, and targeted pretraining. These three pretraining datasets go from focusing on broad vision features, to domain oriented examples, and finally to images that are of our downstream objects of interest. For generalist pretraining, we use model weights created using self-supervised pretraining on ImageNet data. The specialist and targeted pretraining data are described in detail in Section 3.1 and are each generated from the RarePlanes [21] or HRSID [24] dataset and do not include external imagery sources. We do not evaluate representations directly but focus on downstream object detection evaluation compared to the baselines established in RarePlanes and HRSID. We use the VISSL [8] library for model pretraining and Detectron2 [25] library for training and evaluation of object detection models. For our pretraining we used between 8 and 24 V100 GPUs, depending on batch size. Pretraining generally took only a few hours for most experiments but could take up to 8 days when pretraining from scratch. Fine tuning of models was done on either 8 V100's or 16 K80's and took 3 hours and 10 hours to train respectively.

3.1. Datasets

We chose two datasets in the RSI domain that are both focused on detecting objects from overhead, but are very far apart from the visual features available. The RarePlanes dataset is made up of 3 channel RGB images observed from an electro-optical sensor. The HRSID dataset is a single magnitude channel that was collected by a Synthetic Aperture Radar sensor. These types of images are both poorly represented, or not represented at all, in most general pretraining datasets such as ImageNet or CIFAR.

RarePlanes: The RarePlanes dataset [21] is comprised

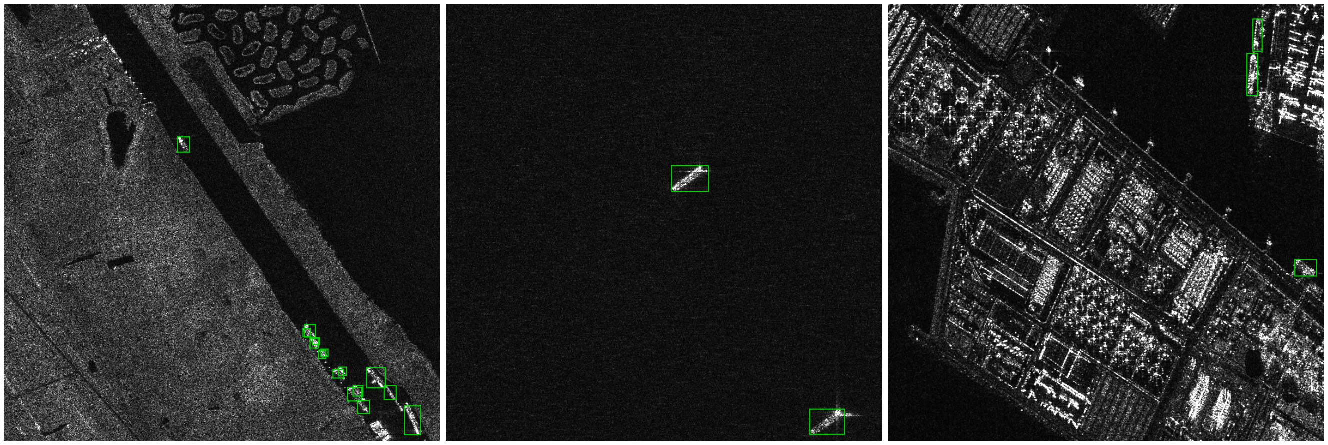


Figure 3: Example of inshore and offshore SAR images in the HRSID dataset. Offshore (center) typically only have objects of interest present in them.

of a real (Figure 1) and synthetic (Figure 2) portion. The real portion is made up of 253 satellite scenes (Maxar WorldView-3 [3]) covering 112 locations. The synthetic portion is comprised of 50k synthetic satellite images with 600k+ aircraft. Although the RarePlanes dataset has 10 different attributes that one can divide the classes amongst, this study simply uses the RarePlanes provided 3 class designations: small, medium, and large.

Model pretraining, supervised fine-tuning, and model evaluation were performed using the established RarePlanes training and evaluation datasets as well as ImageNet pretrained backbones. The dataset was further broken down into specialist and targeted portions for hierarchical pretraining as described in Reed *et al*. [18]. For specialist specific self-supervised pretraining, we created a dataset of 300k unlabeled images that are tiled sections of the large untiled RarePlanes images. These tiled sections are of size 224x224 pixels and are randomly sampled to create smaller subsets of data for specific training experiments. The targeted dataset is a sample of the annotated training chips in the RarePlanes dataset. Using the annotated training chips ensured that each image had at least one object of interest and is therefore ”targeted” to this domain but is still utilized in an unsupervised fashion. Supervised fine-tuning and evaluation was performed using the RarePlanes evaluation dataset. The described method of developing the specialist and targeted datasets applied for both the real and synthetic dataset generation.

HRSID: The HRSID dataset [24] is made up of high resolution SAR images. While the S in SAR denotes ”Synthetic”, this is not the synthetic we are referring to in our paper and instead refers to the satellite's motion, along with advanced signal-processing techniques, to simulate or ”synthesize” a larger antenna. There are 5604 cropped images comprising 16951 ships taken from 136 original panoramic SAR images. The original images were taken from 3 different sensors, Sentinel-1B, TerraSAR-X and TanDEM-X, and all resolutions are under 3m. This dataset also uses 3 classes; small, medium, and large, which are defined by the MS COCO scale division (<32 x 32 pixels, <= 96 x 96 pixels, >96 x 96 pixels). There are a mixture of inshore and offshore scenes as shown in 3, with inshore representing approximately 18% of all scenes. During fine tuning stages with this dataset we resized images to 1000 x 1000 to match the baseline training setup.

We used the established HRSID training and evaluation splits for all experiments the same as the RarePlanes setup. The specialist dataset was developed from a holdout negative dataset provided with the HRSID data, made up of 400 SAR images with no objects. We randomly chipped these into 224x224 crops to generate approximately 25k images for our specialist training. The HRSID targeted dataset was simply all images containing at least one object of interest. We did not filter the data to only inshore or offshore images as Wei *et al*. [24]. However, we provide metrics for, and compare our results to their ResNet50 FasterRCNN as described in more detail below.

3.2. Model

We use the SimCLR [4] framework to pretrain our self-supervised learning representations. SimCLR is a contrastive learning framework that teaches a model to closely associate multiple augmented views of the same image. The SimCLR framework combines stochastic data augmentation, a neural network base encoder, a neural network projection head, and a contrastive loss function, defined as the normalized temperature-scaled cross entropy loss or NT-Xent. This loss function maximizes agreement or similarity,

Dataset	Method	AP	AP-50	AP-Small	AP-Medium	AP-Large
RarePlanes	Shermeyer et al. (Baseline)	68.21	92.10	66.68	70.26	67.68
	ImageNet SSL.	75.06	93.58	71.59	78.83	74.75
	ImageNet → Specialist	75.51	**93.92**	73.15	78.38	**75.00**
	ImageNet → Specialist → Targeted	**75.98**	93.54	**73.45**	**79.64**	74.85
HRSID	Wei et al. (Baseline)	63.5	86.7	64.4	65.1	16.4
	ImageNet SSL.	67.36	90.45	68.28	69.85	35.19
	ImageNet → Specialist	67.91	90.49	**69.05**	**70.37**	39.6
	ImageNet → Specialist → Targeted	**68.03**	**90.58**	69.0	69.14	**42.72**

Table 1: Comparison to HPT object detection metrics compared to baselines provided in Shermeyer *et al.* [21] and Wei *et al.* [24]. Combinations of 3 different HPT stages are shown. Specialist are random samples from imagery and targeted are crops that have at least one known object in them. Details of sampling method are described in Section 3.1

between sampled positive pairs using an adjusted temperature parameter that helps the model learn from hard negatives. The optimal temperature parameter differs with each batch size and number of training epochs. Shown in their original paper, this loss function proved better than alternative loss functions such as logistic loss and margin loss.

We use a ResNet50 with a Feature Pyramid Network backbone for pretraining and model fine-tuning in all experiments. For our downstream object detection network we use Faster-RCNN [19]. This setup is similar to that used in both dataset baselines [21, 24] and allows us to directly compare results.

3.3. Training Setup

We use a hierarchical pretraining setup to conduct our experiments as described in [18]. Hierarchical pretraining involves multiple layers of pretraining to progressively get more similar to the dataset that will be used in the downstream task. All of our experiments begin with a generalist pretrained base model, which is a SimCLR trained ImageNet model from the VISSL model zoo. We then isolate and experiment with different training configurations, e.g. batch size, training iterations, and pretraining stages, to understand where self-supervised pretraining still offers increased performance for small datasets. Due to the tendency for the pretraining model to overfit, we focused many of the pretraining configurations on smaller batch sizes and fewer iterations. Finally, we evaluate by fine-tuning on the training dataset provided in each baseline.

4. Results

Our experiments are consistent with findings from previous research that self-supervised learning produces high-quality visual representations for downstream tasks [4]. Our results specifically demonstrate the potential for these representations to be fine-tuned for object detection using a limited remote sensing dataset. Object detection models fine-tuned from backbones created using self-supervised learning performed better than models fine-tuned from backbones created using supervised learning, whether these self-supervised models were pretrained using ImageNet, domain specialized data, target specific data, or a combination of the 3, as shown in Table 1.

We also find that an optimal batch size for pretraining is smaller when using a narrow domain specific dataset than originally found in [4]. We generally found a negative correlation between batch size and downstream object detection performance with specialized pretraining, indicating that the optimal setup for generalized pretraining (large batch sizes) is different than the optimal setup for specialized pretraining (small batch sizes) with limited target datasets, as shown in Section 4.3. Additionally in Section 4.2, we find differences between the optimal number of training iterations used to pretrain a large generalized base model, as found in [4], as compared to the number of training iterations that are effective for specialized pretraining with limited data. While a base model can be trained for many iterations, we found specialist pretraining to overfit quite quickly, sometimes in fewer than 1,000 training iterations, aligning with results seen in [18].

Finally, we experimented with synthetic data as a sole source of training beyond ImageNet and as an augmentation technique for self-supervised pretraining. These experiments were only performed with the RarePlanes data. We find that synthetic data is useful for specialist pretraining to improve object detection models. We also found self-supervised learning to be less effective in a zero-shot setting, i.e. limited to HPT only on synthetic data, when compared to having unlabeled but domain relevant real data.

4.1. Data Quality Analysis

We analyzed the impact of using a targeted vs specialist set of remote sensing images for fine tuning a base self-supervised pretraining model. For our targeted dataset, we sampled the labelled training data such that each of the images is guaranteed to contain one or more object. Our RarePlanes specialist dataset was created by randomly selecting approximately 5k images from the unlabeled chips generated from methods described in more detail in Section 3.1. This exact number of images was chosen such that the targeted and specialist datasets were the same size and did not bias experiments. As this specialist dataset is created from areas that generally do not contain objects, the diversity is much greater compared to the targeted images which are guaranteed to have at least one object in them. The random images may contain objects such as buildings, roads, planes, or boats, but often only contain grass, dirt, or ocean and are without objects entirely. This is the intrinsic variable object focus that remote sensing imagery can provide for HPT versus ground level imagery which often always has objects of interest in them.

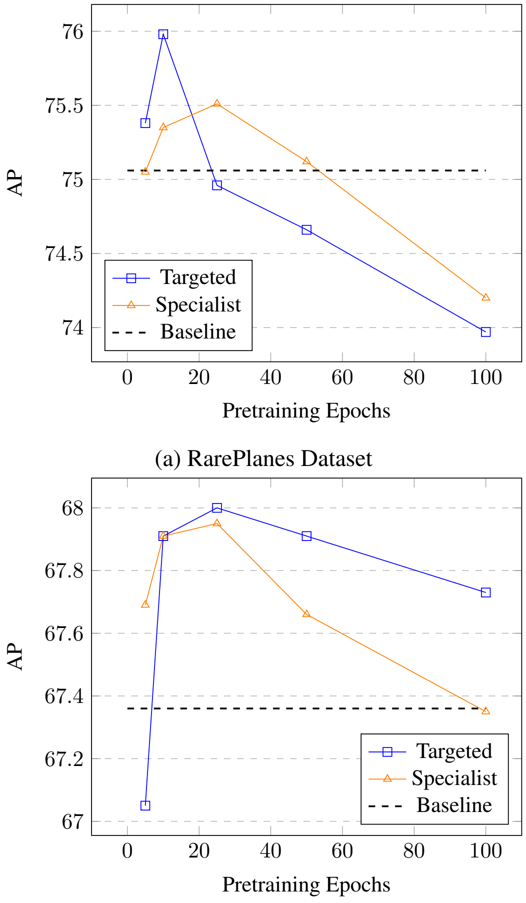
We analyzed the impact of dataset quality across differing number of training iterations. Our results can be found in Figure 4. We found that using the targeted dataset for self-supervised fine tuning led to superior results, indicating that SimCLR may perform better as a pretraining framework when using images with objects. However, using this type of data led to over-fitting more quickly with the RarePlanes data, making the selection of number of training epochs used even more important when using a targeted dataset, but the opposite was found with the HRSID data. In both cases, optimal pretraining epochs were between 10 and 30 which is an observation explored more in the next section.

4.2. Training Length

When using HPT, the generalist pretraining produces high-quality representations that have been shown to outperform supervised methods. We find this to still be the case with our limited dataset, as supervised fine-tuning directly from the base self-supervised weights produces excellent results. When continuing with the other steps of HPT, our experiments have shown that it is easy to overfit when continuing training on the domain-specific datasets, confirming findings from [18].

Figure 4 shows our results when varying the number of training epochs during self-supervised learning. In this experiment, we fine-tune from ImageNet, and experiment with both the targeted and specialized datasets to conduct self-supervised pretraining. We keep batch size, model type, and other values constant.

We find that downstream object detection performance improves with additional self-supervised pretraining from the original base self-supervised weights. However, the model achieves the level of optimal performance early during training before rapidly overfitting to the data. When compared to generalist pretraining as seen in [4], which continues to improve with longer training cycles, the smaller dataset performs in the opposite, which is expected with limited examples in a contrastive learning environment. This can also be shown in the difference between overfitting rates between the targeted and specialist pretraining as discussed in Section 4.1. This difference in overfitting rates highlights the impact that the diversity of data has, which drives how fragile the model is to varying the training length.



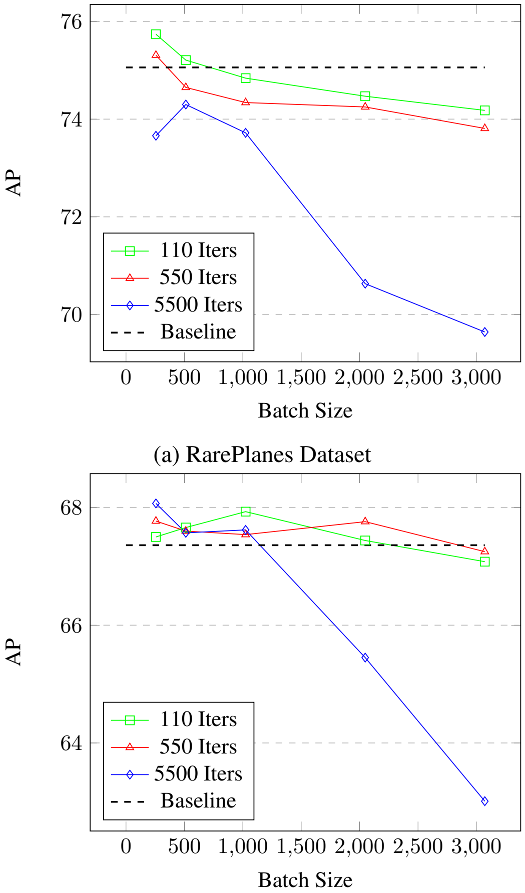
(a) RarePlanes Dataset

(b) HRSID Dataset

Figure 4: Impact that pretraining epochs has on downstream AP score on the test dataset with Targeted and Specialist HPT setups. Targeted data has been sampled such that images have at least one relevant object in them and specialist data is a random sample from all of the original large training images in the training set, which disproportionately do not have relevant objects. The Baseline is the ImageNet self-supervised pretraining.

4.3. Batch Size Analysis

We analyze the impact that batch size selection has on HPT. In [4], the authors find that contrastive learning benefits from larger batch sizes in their experiments using



(a) RarePlanes Dataset

(b) HRSID Dataset

Figure 5: Results of various batch sizes and training iterations has on downstream AP score. Number of training epochs as been adjusted for each batch size so that they roughly align with the displayed iterations; 110, 550, and 5500. This is done so that model update counts are effectively the same for different batch sizes at the same iteration number.

large generalized training datasets. In our experiments, conducted using a smaller, more narrow, domain-specific dataset, we find that smaller batch sizes generally lead to better performance than larger batch sizes with the gap being more consistent the larger the batch size.

We conduct self-supervised pretraining while varying the batch size over three different levels of training iterations. The use of iterations here instead of epochs aligns with the Reed *et al.* [18] method and allows us to ensure that model weight updates are kept consistent for the same batch sizes at different numbers of epochs. We use the targeted datasets for self-supervised training, and start with the ImageNet initialized weights. For supervised fine-tuning and evaluation we use the original train and test datasets. The smallest batch size used was 256, while the largest batch size used was 3,072. Fine-tuned evaluation results are shown in Figure 5.

We find that for lower numbers of iterations the smallest batch size works best and though this shifts slightly for large numbers of iterations the trend of smaller batch sizes are generally better, holds. While Chen *et al.* [4] showed that larger batch sizes with self-supervised learning led to better downstream results, we have seen the opposite relation emerge when the training dataset is much smaller than that of ImageNet. As the batch size starts to become a significant portion of the entire dataset, there becomes less for the model to learn to discriminate from. Additionally, The narrow domain of remote sensing imagery is much more homogenous than generalized vision datasets and discriminating features can be much harder to learn when mixed with many similar examples, as shown in the consistent higher performance of smaller batch sizes.

4.4. Synthetic Data

In addition to the experiments using the RarePlanes real dataset, we explore various self-supervised training scenarios involving the RarePlanes synthetic dataset: 1) Using self-supervised pretraining in a zero-shot setting, where no real labels are used in training, but HPT is done with different stages of synthetic, specialist real data, and both. 2) Leveraging synthetic data in a real data augmentation scenario, where all models are fine tuned on real data and synthetic data is incorporated into some HPT stages. Results can be found in Table 2.

In the zero-shot setting where we have synthetic labelled and real unlabeled data available for training, we found that incorporating the synthetic data into pretraining did not provide improvements over the baseline. While using weights that had been pretrained using generalized and specialist pretraining improved results slightly as compared to the Shermeyer *et al.* [21] fully synthetic experiments.

For the data augmentation scenario, we find that using synthetic data for model pretraining improves downstream object detection performance over the baseline Shermeyer *et al.* [21] fully real experiment results. Although this improvement is in line with improvements of simply performing HPT with the real data alone. We see a drop off in performance when incorporating real and synthetic pretraining but all three scenarios are markedly better than the baseline.

We also performed an experiment to understand the impact of where placing synthetic pretraining in our HPT has on model performance. As seen in Table 2 in the last two rows of each Supervised Training section, where the Specialist → Synthetic and Synthetic → Specialist results are shown. We find that in both fine tuning scenarios having the specialist pretraining as the final pretraining stage leads to generally better improvements. This specialist pretraining

Supervised Training	HPT	AP	AP-50	AP-Small	AP-Medium	AP-Large
Synthetic Fine Tune	Shermeyer et al. (Synth)	35.88	**59.09**	27.70	37.09	42.85
	ImageNet SSL	35.35	57.23	26.46	36.90	42.70
	ImageNet → Synthetic	32.29	53.28	26.20	37.96	32.71
	ImageNet → Specialist	**36.77**	58.00	**28.12**	39.32	**42.87**
	ImageNet → Specialist → Synthetic	33.17	53.02	26.06	37.46	35.98
	ImageNet → Synthetic → Specialist	34.85	56.29	25.89	**39.44**	39.24
Real Fine Tune	Shermeyer et al. (Real)	68.21	92.16	66.68	70.26	67.68
	ImageNet SSL	75.06	93.58	71.59	78.83	74.75
	ImageNet → Synthetic	75.33	**94.13**	71.91	**78.92**	75.16
	ImageNet → Specialist	**75.51**	93.92	**73.15**	78.38	75.00
	ImageNet → Specialist → Synthetic	75.16	93.55	72.48	78.60	74.40
	ImageNet → Synthetic → Specialist	75.44	93.95	72.47	78.50	**75.36**

Table 2: Comparison to HPT object detection metrics compared to baseline provided in Shermeyer *et al*. [21] (all synthetic and all real experiments). Results are split between the supervised training method where **Synthetic Fine Tuning** experiments utilize zero real labels at any training stage and **Real Fine Tuning** uses real labels in the final supervised training stage.

stage aligns more with the downstream evaluation which is also on real data. This difference in performance is more pronounced in the zero shot mode, i.e. the model has supervised fine tuning on synthetic data.

Overall, synthetic data offers more improvement to real data augmentation tasks compared to zero-shot detection tasks when used in pretraining models. Qualitative views of the data, as shown in Figures 1 and 2, show that there are visual features that separate our real and synthetic datasets. Having synthetic data more aligned with the downstream task may close the gap in observed pretraining benefits.

5. Conclusion

In this work we have analyzed the impact a resource constrained dataset has on self-supervised pretraining, how to optimize pretraining for this type of dataset, and how the addition of synthetic data may impact downstream tasks. Our experiments explored how different subtle changes in training configurations affect self-supervised pretraining and show that model performance can still be increased with small amounts of data. We also show that HPT can lead to quick model overfitting when coupled with small datasets and the importance that image diversity plays in being robust to this overfitting.

With regards to remote sensing imagery, we demonstrate that synthetic data can provide a benefit when used in the dataset augmentation role. We also show that the nature of overhead imagery lends itself to HPT tasks as it inherently contains the ability to sample the data in a more targeted way. Taken together, we demonstrate that more niche datasets can still leverage self-supervised pretraining and that synthetic data can provide some benefits in this environment depending the specific training augmentation role.

References

[1] Shekoofeh Azizi, Basil Mustafa, Fiona Ryan, Zach Beaver, Jana von Freyberg, Jonathan Deaton, Aaron Loh, Alan Karthikesalingam, Simon Kornblith, Ting Chen, Vivek Natarajan, and Mohammad Norouzi. Big self-supervised models advance medical image classification. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3458–3468, 2021.

[2] Philip Bachman, R. Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *NeurIPS*, 2019.

[3] Simon Cantrell, Jon Christopherson, Cody Anderson, Gregory L. Stensaas, Shankar N. Ramaseri Chandra, Minsu Kim, and Seonkyung Park. System characterization report on the WorldView-3 Imager. Report 2021-1030I, Reston, VA, 2021. Edition: Version 1.0: June 2021; Version 1.1: October 2021.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020.

[5] Nathan L. Clement, Alan Schoen, Arnold P. Boedihardjo, and Andrew Jenkins. Synthetic data and hierarchical object detection in overhead imagery. *ArXiv*, abs/2102.00103, 2021.

[6] Valéry Dewil, Arnaud Barral, Gabriele Facciolo, and Pablo Arias. Self-supervision versus synthetic datasets: which is the lesser evil in the context of video denoising? In *The 1st Workshop on Vision Datasets Understanding (CVPR 2022)*, New Orleans (Louisiana), United States, June 2022.

[7] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction.

In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1422–1430, 2015.

[8] Priya Goyal, Mathilde Caron, Benjamin Lefaudeux, Min Xu, Pengchao Wang, Vivek Pai, Mannat Singh, Vitaliy Liptchinsky, Ishan Misra, Armand Joulin, and Piotr Bojanowski. Self-supervised pretraining of visual features in the wild. *CoRR*, abs/2103.01988, 2021.

[9] Julius Von Kgelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. Self-supervised learning with data augmentations provably isolates content from style. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[10] Benjamin Kiefer, David Ott, and Andreas Zell. Leveraging synthetic data in object detection on unmanned aerial vehicles. *ArXiv*, abs/2112.12252, 2021.

[11] Fanjie Kong, Bohao Huang, Kyle Bradbury, and Jordan M. Malof. The synthinel-1 dataset: a collection of high resolution synthetic overhead imagery for building segmentation. *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1803–1812, 2020.

[12] Weixing Liu, Bin Luo, and Jun Liu. Synthetic data augmentation using multiscale attention cyclegan for aircraft detection in remote sensing images. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022.

[13] Charles Thane MacKay and Teng-Sheng Moh. Learning for free: Object detectors trained on synthetic data. In *2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, pages 1–8, 2021.

[14] Basil Mustafa, Aaron Loh, Jan Freyberg, Patricia MacWilliams, Megan Wilson, Scott Mayer McKinney, Marcin Sieniek, Jim Winkens, Yuan Liu, Peggy Bui, et al. Supervised transfer learning at scale for medical imaging. *arXiv preprint arXiv:2101.05913*, 2021.

[15] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 512–523. Curran Associates, Inc., 2020.

[16] Michael D. Presnar, John P. Kerekes, and David R. Pogorzala. Dynamic image simulations for adaptive sensor performance predictions. In *2010 2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, pages 1–4, 2010.

[17] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022.

[18] Colorado Reed, Xiangyu Yue, Aniruddha Nrusimha, Sayna Ebrahimi, Vivek Vijaykumar, Richard Mao, Bo Li, Shanghang Zhang, Devin Guillory, Sean Metzger, Kurt Keutzer, and Trevor Darrell. Self-supervised pretraining improves self-supervised pretraining. *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1050–1060, 2022.

[19] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015.

[20] Karsten Schulz, Ronny Hänsch, and Uwe Sörgel. Machine learning methods for remote sensing applications: an overview. In Ulrich Michel and Karsten Schulz, editors, *Earth Resources and Environmental Remote Sensing/GIS Applications IX*, volume 10790, pages 1 – 11. International Society for Optics and Photonics, SPIE, 2018.

[21] Jacob Shermeyer, Thomas Hossler, Adam Van Etten, Daniel Hogan, Ryan Lewis, and Daeil Kim. Rareplanes: Synthetic data takes flight. *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 207–217, 2021.

[22] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2242–2251, 2017.

[23] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *ArXiv*, abs/1807.03748, 2018.

[24] Shunjun Wei, Xiangfeng Zeng, Qizhe Qu, Mou Wang, Hao Su, and Jun Shi. Hrsid: A high-resolution sar images dataset for ship detection and instance segmentation. *IEEE Access*, 8:120234–120254, 2020.

[25] Y. Wu, F. Kirillov, F. Mass, W.-Y. Low, and R. Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.