Beyond output-mask comparison: A self-supervised inspired object scoring system for building change detection

Are C. Jensen

Norwegian Computing Center (NR)
arej@nr.no

Abstract

Updating urban-area maps is crucial for urban planning and development. Traditional methods of updating urban-area maps based on aerial photography are labor-intensive and struggle to keep pace with rapid urban development. Automated algorithms for detecting new and removed buildings based on bi-temporal images typically either rely on comparing mono-temporal building detection outputs or requiring examples of new and removed buildings for training. This study presents a novel method using self-supervised learning principles to train a distinct object-change scoring network. It repurposes segments of the (potentially imperfect) delineations used in single-temporal detector training, harnesses bi-temporal data attributes, and leverages the assumption that most buildings remain unchanged over time. This eliminates the need for explicit examples of new or removed buildings, while still overcome usual constraints of post-detection output-mask comparison methods. We provide precision-recall curves and examples demonstrating the improved performance of the suggested approach. Furthermore, we discuss several immediate algorithmic variations that hold the potential for even further enhancements in performance.

1 Introduction

Urban areas are continuously evolving to accommodate growing populations, technological advancements, and socio-economic demands. Accurate and up-to-date maps are essential for urban planning, infrastructure development, emergency response, and real estate management. Traditionally, updating urban-area maps based on aerial photography has been labor-intensive and time-consuming. Although more involved image products, like stereo-imaging, LIDAR or 3D renditions, can provide significant information about changes in building structure, simpler orthorectified bi-temporal aerial images are often more readily available and can also provide valuable insights into building changes. Automating the analysis of such images can thus diminish the necessity for human labor and/or mitigate errors stemming from human involvement.

This study focuses on analyzing orthorectified images captured at two distinct time points, T0 (the year 2018) and T1 (2020). Our objective is to identify buildings that have been constructed, altered, or removed during this interval. We do, however have access to (potentially imperfect) delineations at T0, providing us with a form of ground truth for that time point. These delineations can be used to train mono-temporal building detection algorithms. While advancements in such mono-temporal detectors have been made, they inherently face limitations as change detectors when applied separately to T0 and T1 images. A common alternative is to train a system to detect changes directly, but this requires annotated examples of new, removed, and altered buildings.

To tackle these challenges, we propose a novel method grounded in self-supervised learning. Our strategy involves deploying a distinct scoring network that assesses each detected building, assigning a score that ideally reflects the likelihood of structural changes. This network is trained by repurposing the delineations from the single-temporal detector training, harnessing the characteristics of bi-temporal data, and building on the premise that most buildings remain unchanged over time. This methodology allows the network to learn from the variances and consistencies observed in the building structures across different time frames.

The rest of the paper is organized as follows: Section 2 discusses related work in the field of building change detection and self-supervised learning. Section 3 describes the dataset and its specifics. Section 4 details the proposed scoring network and experimental setup. Section 5 presents the results and discussion, as well as outlines potential algorithmic variations for further improvement. Finally, Section 6 concludes the paper.

2 Related work

Various methodologies for automated building change detection have been proposed, however methodologies based on deep learning, as in almost all other image analysis areas, have become the dominant approach [1].

Proceedings of the 5th Northern Lights Deep Learning Conference (NLDL), PMLR 233, 2024.

2.1 Output mask comparison

A direct and seemingly intuitive approach to detect changes in buildings is to compare the output masks generated by building detection algorithms. Typical algorithms are variations and more modern derivatives of the U-net [2] or Mask-RCNN [3] architectures. Note that these detectors are applied on mono-temporal images, and hence their training is not dependent on actual examples of temporal changes in building structures. Groups of pixels for the former single-pixel segmentation approaches, or the object-level instances produced by the latter are often the preferred level of granularity for detecting change [1]. By analyzing the differences between masks from two time points, T0 and T1, one can theoretically pinpoint new constructions, modifications, and demolitions. However, this method has shown to be less than ideal for several reasons, among them:

Spatial overlap: Urban redevelopment often sees new buildings constructed on or near the sites of older structures. This spatial overlap can lead to confusion in simple mask comparison methods.

False positives: To ensure comprehensive building detection, the threshold is often set low. This results in a high number of false positives inconsistently over T0 and T1, causing false detections.

Ambiguities: In many instances, buildings may be partly obscured by elements like vegetation or may not appear distinctly in the aerial imagery. This necessitates a more direct comparison of the T0 and T1 images themselves.

Masking inconsistencies: Small input variations, like the differences seen in T0 and T1 images can produce varying output masks. For instance, one might merge adjacent smaller buildings, or perhaps split larger structures into multiple entities.

Examples can be found in Figure 2 of the Data and dataset section.

2.2 Bi-temporal data training

An alternative and well-established approach is to train a deep learning network specifically for detecting new and removed buildings using bi-temporal data as input. One immediate approach is of course to concatenate the T0 and T1 images and use them as input to existing detection or segmentation network architectures, e.g. [4]. A more common approach is to send the T0 and T1 image pair separately through an encoder network for extracting features, while a temporal feature difference decoder detects object change, like in the FC-Siam network [5] and derivatives (see [1] and references therein). Even though this approach is conceptually appealing, has great potential and many structural varieties have been studied, there is still a need for labeled training data. For effective network training, numerous examples of new, altered, and removed buildings are required. While some techniques, such as the one employed in "ChangeStar" [6], have been developed to artificially generate such data, they often simplify the problem by dichotomizing it into building-no-building. This binary perspective fails to account for nuanced scenarios where a building has been replaced or partially replaced by new construction.

2.3 Self-supervised learning and the VICReg algorithm

Unlike traditional supervised methods that rely heavily on labeled data, self-supervised algorithms like VICReg [7] leverage unlabeled data to learn meaningful representations. The core principle behind VICReg, and many other self-supervised approaches, is to aim for consistent representations (or embeddings) of images with similar semantic content. This consistency ensures that imaged objects, irrespective of minor changes or occlusions, are represented similarly in the embedding space, making it particularly relevant for tasks like building change detection. The challenge is to artificially create or naturally bring about image variants with similar semantic content for training. Applying this technique directly as a pixel-level change detector has seen potential [8], although when applying it as a pre-training step in the previously-mentioned algorithms one still needs examples of actual changes for training the final system.

In the context of this study, we explore the potential of the VICReg algorithm to train a separate deep learning network. This network, when combined with existing building detection algorithms, aims to provide a more nuanced and accurate scoring system to assess the likelihood of structural changes in buildings between T0 and T1 without the need for actual examples of changes to train on. See section 4 for more details.

3 Data and dataset specifics

The dataset under consideration comprises bi-temporal orthorectified aerial images captured in two distinct years: T0 (2018) and T1 (2020). These images have been generously provided by Field (www.field.group). The dataset offers a comprehensive view of partly urban landscapes, capturing the dynamic nature of building constructions, alterations, and demolitions over the two-year period.

The aerial images have a high spatial resolution of 10 cm by 10 cm, ensuring detailed and clear representations of urban structures. Example images can be found in Figure 1. In total, the dataset encompasses approximately 65,000 buildings, represented in around 40,000 RGB images, each of size 1500x2000 pixels.

While the images are comprehensive and having a high spatial resolution, they are not without their challenges:

Alignment issues: A notable challenge is the lack of pixel-perfect alignment between T0 and T1 images. This misalignment can be attributed mainly to the inaccuracies or limitations in the orthorectification process. Differences in camera angles during the two aerial captures exacerbate this issue.

Natural variations: The process of detecting changes in aerial imagery is often complicated by natural variations over time. Variations in lighting, influenced by the time of day or weather conditions during image capture, can create inconsistencies. Seasonal changes, such as the growth or loss of vegetation, may obscure or alter how buildings appear. Additionally, factors like the aging of buildings, which leads to visual changes, and activities unrelated to building construction, can further contribute to discrepancies.

Accompanying the aerial images for the year T0 is a dataset that attempts to outline the building structures; however, it is crucial to note that this "ground truth" is not without its flaws. Not all buildings present in the images are marked, and some delineations indicate planned buildings not yet materialized. In addition, the delineations are not pixel perfect, cf. the general challenges and impediments mentioned above.

For validation and testing purposes, a separate test set has been curated. This set consists of 40 images, each meticulously hand-crafted to delineate new and removed buildings. This test set will serve as a benchmark to evaluate the performance and accuracy of the proposed building change detection methodology.

Figure 1. Example images from the high-resolution bi-temporal dataset. T0 has building delineations, although flawed, avilable, as shown superimposed.

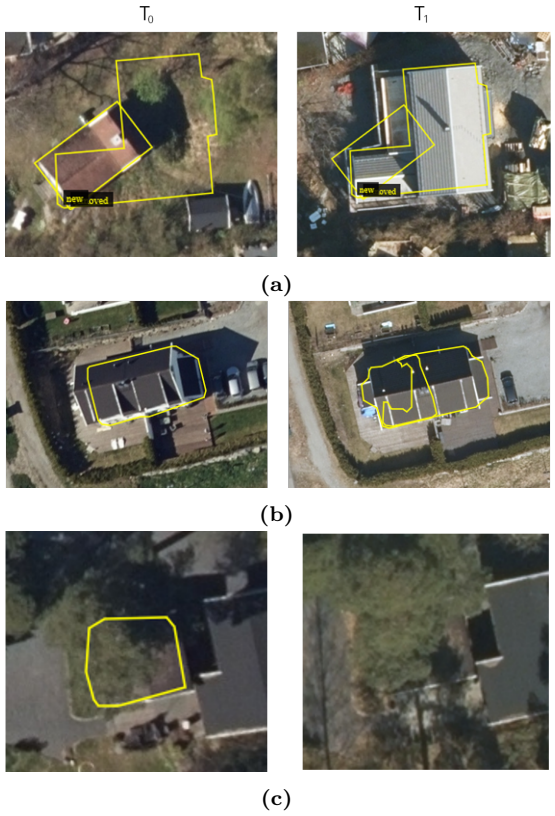

Figure 2. Challenges in post-detection mask comparisons: a) An instance showcasing spatial overlap between a new and a removed building. b) A demonstration of T0 and T1 output mask inconsistency. c) A scenario where discerning the building structure proves challenging due to external factors, here vegetation.

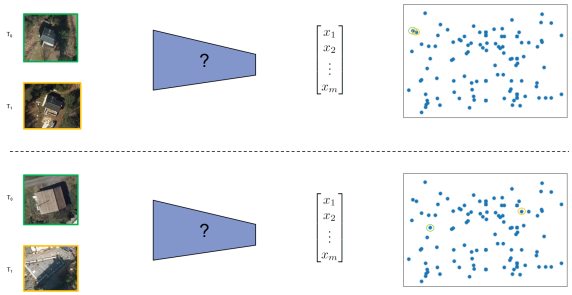

Figure 3. A conceptual illustration showcasing the scoring network's ideal functionality. Embeddings are closely aligned when a building persists between T0 and T1, while they diverge significantly when a building from T0 is absent in T1.

4 Scoring network and experimental setup

4.1 Overview

The core of our proposed methodology revolves around the utilization of a separate network designed to map building crops into a vector space. In this space, the distances between vectors should

be indicative of the likelihood or score of change between two time points, T0 and T1. The setup and training of the network is inspired by the principles of self-supervised learning.

It's important to clarify that we assume that we already have a trained building detector. In our case a Mask-RCNN with a ResNet-101 [9] feature pyramid network [10] backbone. What is described here is the preparation and training of the separate network which scores the building detection candidates from either of the T0 or T1 images. A conceptual illustration of the intended operation of such a network can be found in Figure 3.

4.2 Data preparation and assumptions

For the training process, we utilize crops of buildings from both T0 and T1, based on the (although inaccurate) ground truth delineations available for T0. A fundamental assumption underpinning our approach is the persistence of most buildings over time. That is, we treat these T0-T1 patches as if they each were depicting the same, unaltered building in both T0 and T1. The dataset provides approximately 65,000 T0-T1 crop pairs.

To enhance the generalizability and robustness of the network, we introduce several data augmentations. These include slight rotations, resizing, blurring, and color jittering. That is, we obtain different depictions of the same building through both natural variations as we progress along the temporal axis and artificial modifications achieved through augmentations. Finally, the patches are resized to a fixed dimension of 224x224 pixels and undergo spatial tapering before being fed into the network.

4.3 Network architecture and loss function

For the network architecture, we employ a ResNet-50, turning (embedding) the image patches into vectors, $\mathbf{x}$, of length 2048. The chosen loss function combines the squared Euclidean distance between the embeddings of T0 and T1 crops, $\mathbf{x}_{T0}$ and $\mathbf{x}_{T1}$, respectively, and a term that encourages diversity in the embeddings to avoid a trivial solution, akin to that of VICReg:

$$L = ||\mathbf{x}_{T0} - \mathbf{x}_{T1}||^2 + \text{non-spherical spread of } \mathbf{x}. \quad (1)$$

The second term in (1) is based on calculating the empirical covariance matrices of $\mathbf{x}_{T0}$ and $\mathbf{x}_{T1}$, respectively, and penalizing non-diagonals as well as diagonals having less than unit value. More precisely, we sum the square of the non-diagonals, and sum the square of 1 minus the diagonals for diagonals having a value less than one. These two terms are added together, although with the former having a weight 1/25 as suggested in [7]. Please refer to the paper just cited for exact details.

4.4 Computational constraints

Given the computational intensity of deep learning tasks, it is worth noting that our training was conducted on a single NVIDIA RTX 2080-Ti GPU, spanning approximately 24 hours. The starting point was the PyTorch's IMAGENET1K_V1 pre-trained weights.

4.5 Exploration with foundation models

In addition to the primary approach, we also explored the potential of fully self-supervised foundation models, notably CLIP [11] and DINOv2 (ViT-B) [12]. These models, applied out-of-the-box without any fine-tuning, were tested for their ability to meaningfully embed the building crops. We experimented with both Euclidean distances and cosine similarity as scoring functions, however they provided similar results in our setting.

Details related to an attempt at finetuning the DINOv2 model by the exact same setup as when training the ResNet-50 can be found in the Appendix.

4.6 Output-mask comparison scoring

When comparing output masks directly, without the separate scoring network, we calculate a change-score using the following: For objects detected in T0, we first weight the output mask by its detection-score value, and then calculate the mean pixel-wise difference for this mask to a cumulative detection output-mask for T1. Similarly, although reversed, for detections in T1.

5 Results and discussion

The primary objective of our study was to evaluate the efficacy of the proposed VICReg-inspired approach in scoring potential changes in buildings between two distinct time points, T0 and T1. The precision-recall curves, depicted in Figure 5, and the derived average precision scores (APs), summarized in Table 1, collectively indicate that the suggested approach provides meaningful trade-offs between precision and recall for both detecting new and removed buildings.

Figure 4 presents visual examples of T0-T1 building-crop pairs and their corresponding score values. It is evident that the scoring mechanism accurately reflects the structural changes between T0 and T1 in most cases. However, there are instances

where the scoring mechanism fails. These failures seldom occur, though, when the system produces a low score, which is associated with a low likelihood of change.

The out-of-the-box pre-trained foundation models like DINOv2 did not exhibit good performance. Results are not shown, but CLIP performed even worse than DINOv2. This underscores the importance of training the network on domain-specific data, as the general features learned by foundation models like DINOv2 may not be directly transferable to specialized tasks such as building change detection. It can be mentioned that an initial attempt at finetuning the DINOv2 using the exact same setup as when training the ResNet-50 scoring network provided only minuscule improvements. Please refer to the Appendix for further details.

Overall, the results indicate that the suggested VICReg-inspired approach effectively addresses the limitations of the traditional post-mask comparison method. By incorporating either the raw pixel values or a set of "deeper" features from the images, our method offers a more comprehensive and accurate assessment of whether a building persists over time.

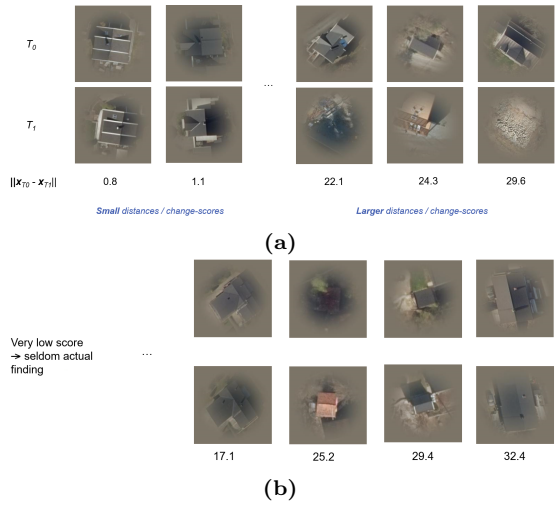

Figure 4. a) Visual representation of T0-T1 building crop pairs alongside their score values, showcasing the accuracy of the suggested approach. b) Instances where the scoring mechanism fails, though these are rare occurrences when the score is low.

Method	New (buildings)	Removed
Post-detection	0.75	0.58
Suggested	0.80	0.67
DINOv2	0.54	0.38

Table 1. Summarizing the precision-recall curves in Figure 5 as average precisions scores (APs). Note that the DINOv2 is used here as an embedder without any finetuning.

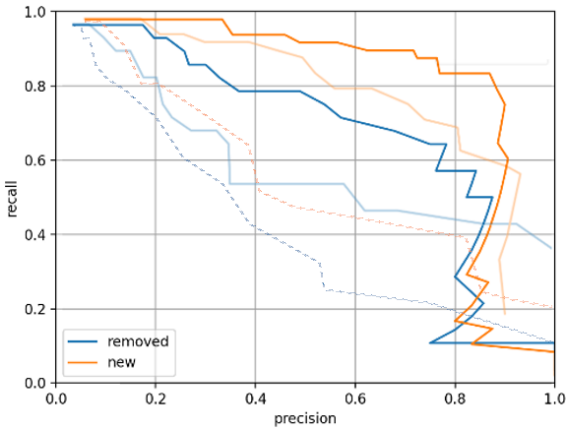

Figure 5. Precision-recall curves juxtaposing the performance of the suggested approach (highlighted in solid) against the traditional post-detection algorithm (depicted faintly) and the out-of-the-box DINOv2 embedding using Euclidean distances (shown dashed). Note that we report detections of new and removed buildings separately.

5.1 Immediate algorithmic variations

The results of our study are promising, but there are several potential variations and enhancements that could be explored to further optimize the performance of the system:

- **Application-specific augmentations:** While we have used common augmentations like rotation, resizing, blurring, and color jittering, there might be room for improvement by incorporating more application-specific augmentations. For instance, augmentations that simulate common challenges in aerial imagery, such as varying lighting conditions, seasonal changes, or occlusions due to vegetation, could make the model more robust to real-world variations.

- **Context-dependent tapering:** We currently use context-independent tapering before feeding patches into the network. A potential improvement could be replacing this with tapering related to the spatial detection mask.

- **Consistent input resolution:** At present, building patches are resized to 224 x 224 pixels for analysis, irrespective of their original size. Maintaining the original ground sampling rate consistently, which also typically offers higher resolution, may improve the network's change detection accuracy.

- **Link to building detection network:** Our approach treats the change-scoring network as a separate entity from the building detection network. A potential variation could be to link the building detection network to the change-

scoring network by e.g. using its backbone as a starting point for training.

- **More or fully unsupervised:** Our current approach relies on (slightly inaccurate) T0 ground truth data for training the scoring network. An interesting variation would be to explore utilizing more of the data, or move towards a fully unsupervised approach, where the training of the scoring model does not depend on any ground truth data at all.
- **Exploration of alternative loss functions:** The loss function we currently employ focuses solely on similarity and penalizing non-spread. Investigating different loss functions, particularly those incorporating contrastive elements, could potentially offer practical advantages.
- **Refined strategies for foundation model utilization:** Despite the suboptimal performance of our initial attempt at fine-tuning DINOv2, detailed in the Appendix, the potential benefits of leveraging large pre-trained models remain significant. With appropriate adjustments and a deeper understanding of how to harness the extensive knowledge embedded in these models, there is a possibility of achieving enhanced performance.

6 Conclusion

This study underscores the inadequacy of relying solely on the analysis and merging of information from two mono-temporal building detections to identify changes. It is evident that the images themselves or a set of "deeper" features must be incorporated into the process to accurately predict whether a building persists over time.

Our proposed approach, leveraging self-supervised learning principles to train a separate object scoring network, has demonstrated promising results. The scoring network is applied in conjunction with an already-trained mono-temporal building detector. The approach circumvents the need for explicit examples of new and removed buildings, which are often challenging to obtain in sufficient quantities. Instead, it repurposes portions of the (potentially imperfect) delineations used for training the single-temporal detector, capitalizes on the bi-temporal characteristics of the data, and leverages the presumption that most buildings remain unaltered over time.

While we have shown advancements over the simpler methods of comparing mono-temporal detection outputs, we have also discussed algorithmic variations that could offer additional performance improvements.

References

[1] H. Jiang, M. Peng, Y. Zhong, H. Xie, Z. Hao, J. Lin, X. Ma, and X. Hu. "A Survey on Deep Learning-Based Change Detection from High-Resolution Remote Sensing Images". In: *Remote Sensing* 14.7 (2022). DOI: 10.3390/rs14071552.

[2] O. Ronneberger, P. Fischer, and T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: vol. 9351. Oct. 2015, pp. 234–241. ISBN: 978-3-319-24573-7. DOI: 10.1007/978-3-319-24574-4_28.

[3] K. He, G. Gkioxari, P. Dollár, and R. Girshick. "Mask R-CNN". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2980–2988. DOI: 10.1109/ICCV.2017.322.

[4] Q. Wang, X. Zhang, G. Chen, F. Dai, Y. Gong, and K. Zhu. "Change detection based on Faster R-CNN for high-resolution remote sensing images". In: *Remote sensing letters* 9.10 (2018), pp. 923–932.

[5] R. Daudt, B. Saux, and A. Boulch. "Fully Convolutional Siamese Networks for Change Detection". In: Oct. 2018, pp. 4063–4067. DOI: 10.1109/ICIP.2018.8451652.

[6] Z. Zheng, A. Ma, L. Zhang, and Y. Zhong. "Change is Everywhere: Single-Temporal Supervised Object Change Detection in Remote Sensing Imagery". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 15193–15202.

[7] A. Bardes, J. Ponce, and Y. LeCun. "VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning". In: *International Conference on Learning Representations*. 2022.

[8] Y. Chen and L. Bruzzone. "Self-Supervised Change Detection in Multiview Remote Sensing Images". In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), pp. 1–12. DOI: 10.1109/TGRS.2021.3089453.

[9] K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[10] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. "Feature Pyramid Networks for Object Detection". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 936–944. DOI: 10.1109/CVPR.2017.106.

[11] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. "Learning Transferable Visual Models From Natural Language Supervision". In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. 18–24 Jul 2021, pp. 8748–8763.

[12] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, R. Howes, P.-Y. Huang, H. Xu, V. Sharma, S.-W. Li, W. Galuba, M. Rabbat, M. Assran, N. Ballas, G. Synnaeve, I. Misra, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski. *DINOv2: Learning Robust Visual Features without Supervision*. 2023.

A Initial attempt at finetuning DINOv2 as a scoring network

In an attempt to enhance our building-change scoring framework, we replaced the ResNet-50 with DINOv2, using the same training patches, loss function, and computational resources. However, the results were underwhelming: the average precision (AP) for detecting new buildings was 0.60, and for removed buildings, 0.48. These outcomes slightly surpassed the baseline performance of DINOv2 but fell short of our ResNet-50 based model.

Possible reasons for this discrepancy include the higher computational demands of DINOv2, the misalignment of our image preprocessing methods with DINOv2's pre-training, a deviation between the loss functions used in DINOv2's pre-training and its current optimization, the reduced-dimensional embedding of DINOv2 (768) compared to that of ResNet-50 (2048), the destabilizing impact on (1) due to the smaller batch size required because of the increased memory demands of the model, and the use of non-optimal hyperparameters for training this specific model.

These insights suggest that effectively leveraging such advanced models requires more specialized adaptations to suit the unique challenges of our application.