Self-Supervised Learning for Real-World Object Detection: a Survey

Alina Ciocarlan, Sidonie Lefebvre, Sylvie Le Hégarat-Mascle and Arnaud Woiselle

arXiv:2410.07442v2 [cs.CV] 11 Oct 2024

***Abstract*—Self-Supervised Learning (SSL) has emerged as a promising approach in computer vision, enabling networks to learn meaningful representations from large unlabeled datasets. SSL methods fall into two main categories: instance discrimination and Masked Image Modeling (MIM). While instance discrimination is fundamental to SSL, it was originally designed for classification and may be less effective for object detection, particularly for small objects. In this survey, we focus on SSL methods specifically tailored for real-world object detection, with an emphasis on detecting small objects in complex environments. Unlike previous surveys, we offer a detailed comparison of SSL strategies, including object-level instance discrimination and MIM methods, and assess their effectiveness for small object detection using both CNN and ViT-based architectures. Specifically, our benchmark is performed on the widely-used COCO dataset, as well as on a specialized real-world dataset focused on vehicle detection in infrared remote sensing imagery. We also assess the impact of pre-training on custom domain-specific datasets, highlighting how certain SSL strategies are better suited for handling uncurated data.**

Our findings highlight that instance discrimination methods perform well with CNN-based encoders, while MIM methods are better suited for ViT-based architectures and custom dataset pre-training. This survey provides a practical guide for selecting optimal SSL strategies, taking into account factors such as backbone architecture, object size, and custom pre-training requirements. Ultimately, we show that choosing an appropriate SSL pre-training strategy, along with a suitable encoder, significantly enhances performance in real-world object detection, particularly for small object detection in frugal settings.

***Index Terms*—Self-supervised learning, small object detection, domain-specific pre-training, frugal setting.**

I. Introduction

Self-supervised learning (SSL) is an exciting and active research area in computer vision. It consists in an unsupervised training of deep learning networks (often only the encoder) using a well-designed pretext task. The aim of this pre-training task is to help the network learning features or invariances that are relevant for the downstream task. In the literature, SSL methods have been shown to improve SOTA performance for many use cases. More specifically, SSL allows the network to learn general features from large unlabelled datasets which, when transferred to a final task, will improve performance despite difficult fine-tuning conditions (e.g., little annotated data or few computational resources).

Fundamental SSL methods deal with instance discrimination, which aims at modeling the decision borders between sub-sets of data represented in the latent space. These methods consider images as instances, and perform inter-image discrimination. Concretely, the optimization aims to minimize, in the latent space, the distance between features of instances that share similar semantic properties (e.g., augmented views from the same anchor images). Emblematic methods include MoCov2 [3], BYOL [4] and DINO [5]. We refer the reader to the following surveys [6], [7], [8] for more details about instance discrimination methods.

However, instance discrimination methods were primarily designed for classification tasks, and most of them are benchmarked on classification datasets only. Although some methods [3], [9] provide promising results on famous object detection datasets like COCO [1] or ADE20K [10], they were not specifically designed for object detection and thus may appear sub-optimal for this task, and even worse for small object detection. This is especially true for instance discrimination methods that mostly

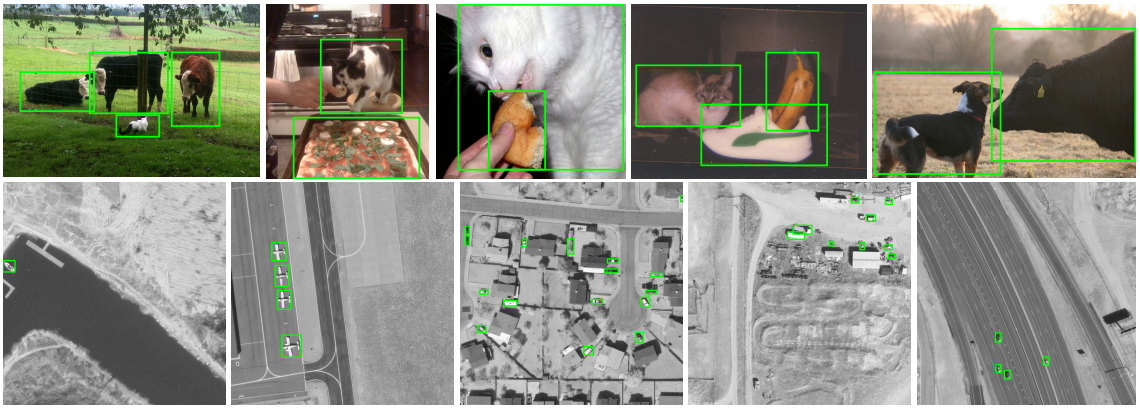

Fig. 1: Example of images dealing with object detection. The first row shows some images taken from the COCO dataset [1], and the second row provides some infrared images taken from the VEDAI dataset [2]. Objects are framed in green.

involve inter-image comparisons, assuming that the images are semantically consistent. To overcome this problem, some object-level instance discrimination methods have been proposed in the literature [11], [12], [13], [14], [15], [16]. They either rely on local crops to create positive pairs, or on dense instance discrimination loss. Then, another recently introduced SSL paradigm that deals with local feature analysis is Masked Image Modeling (MIM). Unlike instance discrimination, MIM methods naturally deal with modeling local relationships: neighboring pixels are all the more important to reconstruct masked patches.

Although object-level instance discrimination and MIM methods have been shown to be efficient for local or dense prediction tasks, it remains unclear which paradigm is better suited for object detection. Few studies have attempted to compare instance discrimination and MIM paradigms [17], [18], [19], [20]. These studies all agree that MIM methods lead to better performance than instance discrimination methods when fine-tuned on data-sufficient object detection dataset. Specifically, the authors of [19] observe that MIM shows a local inductive bias at all layers while MoCov3 (that is an instance discrimination method) tends to focus on local details in lower layers and on global details in higher layers. They also show that MIM pre-training brings sufficient diversity to the attention heads, unlike instance discrimination pre-training strategies whose capacity may thus be limited. The authors conclude that coupling MIM methods with Vision Transformer (ViT) encoders should lead to state-of-the-art (SOTA) performance. However, [20] extends these studies by evaluating these methods in data-limited contexts, and concludes that while MIM methods often outperform contrastive learning methods on large downstream datasets, they struggle with smaller datasets.

These surveys have two key limitations: 1) they do not consider object-level instance discrimination methods, and 2) they rely exclusively on ViT backbones. As a result, the conclusions have not been validated on CNN-based backbones such as ResNets. CNN-based encoders remain widely used in many real-world applications and have some advantages, such as faster inference times, and a hierarchical architecture that benefits object detection. The authors of [21] evaluate some local instance discrimination methods using a ResNet-50 backbone, but they only considered a few-shot setting and did not compare with MIM methods. Moreover, the results were directly taken from the original papers, which, as the authors noted, could lead to unfair comparisons due to differences in implementation.

We aim to address these gaps by providing a

comprehensive survey of SSL methods tailored for object detection, with a focus on challenging cases such as small objects or frugal contexts. Specifically, we extensively cover local instance discrimination and MIM methods. We then benchmark a selection of methods, ensuring the representativeness across all SSL categories and network architectures. We compare global, local instance discrimination, and MIM methods using two network sizes, considering both ResNet-50 and ViT backbones. Our first benchmark uses the widely recognized COCO dataset, with a particular focus on the performance on small objects. We then move to a real-world application involving small object detection, namely vehicle detection from remote sensing data, using the VEDAI dataset [2]. Some examples of images taken from both datasets are provided in Figure 1. An important limitation of previous studies is their primary focus on the COCO dataset, which is not representative of many real-world object detection scenarios. In practical applications, objects may be very different (e.g., very small) and hidden within complex backgrounds. Additionally, different sensors may be used, such as hyperspectral sensors, making pre-trained weights on RGB images less applicable. In such cases, it is necessary to train SSL methods on a dataset that shares similar spectral characteristics with the target dataset. The quality of SSL pre-training (i.e., pre-training that leads to high fine-tuning performance) and the choice of SSL method will then heavily depend on the characteristics of the pre-training dataset (e.g., temporal redundancy, image diversity, dataset size, etc.). We therefore propose an experiment where we pre-train on a large-scale, non-curated IR dataset and we evaluate the benefits of custom pre-training on the IR version of the VEDAI dataset compared to using weights pre-trained on RGB images.

Our contributions can be summarized as follows:

- We provide an exhaustive survey of SSL methods tailored for object detection, with a specific focus on local instance discrimination and MIM methods.
- We evaluate representative SSL methods from each category on two benchmarks. First, we consider the widely used COCO dataset, emphasizing metrics related to small objects. Then, we evaluate these SSL strategies in a real-world application, specifically vehicle detection from remote sensing data. This allows us to draw conclusions on the optimal SSL strategy depending on various parameters (ResNet or ViT backbone, object size, fine-tuning dataset size, etc.).
- We offer insights on which SSL strategy to use when pre-training on an in-domain dataset is required.

II. Towards local-level self-supervised learning

In this section, we present some SSL strategies that are better suited to dense or local prediction tasks (e.g., segmentation and object detection, respectively) as they aim to learn local features. They can be grouped within two categories, namely object-level instance discrimination methods and masked image modeling. Table I summarizes the different categories and the associated methods that will be discussed.

A. Object-level instance discrimination methods

Some authors proposed variants of instance discrimination methods that are well suited to object detection tasks. Instance discrimination methods aim at minimizing the distance in the latent space between features of instances that share similar semantic properties. The fundamental methods presented in the Introduction perform only inter-image comparisons: they generally consider the entire images as their instances, assuming that these are semantically consistent. This is indeed the case when the methods are trained on object-centric datasets such as ImageNet. However, this hypothesis does not necessarily hold when dealing with dense prediction tasks such as object detection or segmentation. To overcome this issue, two approaches have been investigated: designing data-augmentations at the object or region-level, or applying instance discrimination loss at a local-level (e.g., per pixel).

Object-level instance discrimination	*Region-level augmentations*		SCRL, **ReSim**, MaskCo, SoCo, CAST, ContrastiveCrop, InsLoc, CP², ORL, **Leopart**, InsCon
	Dense loss	Raw pixels	VaDeR, PixContrast, PixPro, DUPR, InsCon, **Leopart**, LC-Loss, CLOVE
		Feature matching	DenseCL, Self-EMD, VicRegL
		Semantic alignment	DetCon, Odin, SetSim
Masked Image Modeling	*Specific masking strategy*		CIM, MST, AttMask, AMT, MILAN, SemMAE, DPPMask, MixMAE
	Target objective	Geometric alignment	**MAE**, SimMIM, ConvMAE, **SparK**
		Image descriptors	PixMIM, Ge²-AE, A²MIM, MaskFeat, SSM
		Deep features	BEiT, MaskDistill, MILAN, MaskAlign, SplitMask, iBOT, I-JEPA, dBOT

TABLE I: Taxonomy of local-level SSL methods for image representation learning. The methods we will consider in our experiments are shown in bold.

1) Region-level augmentations: The approach consists in applying instance discrimination loss to local patches in order to perform intra-image instance discrimination. Several strategies have been proposed to ensure semantic consistency between images that form a positive pair. Spatially Consistent Representation Learning (SCRL) [22] first proposed to randomly select boxes within the intersecting area of the two positive samples and to minimize the similarity between the features predicted by the pooled boxes. Concurrently, [11] proposed a similar approach called ReSim. As shown in Figure 2, a sliding window extracts, in each branch, local features within the overlapping area between the two augmented views of the anchor sample (dashed green area). This creates local positive pairs that represent exactly the same spatial region in the original image (we say that the patches are geometrically aligned). Unlike SCRL, the loss is applied at three different scales in the network, which benefits the detection of objects of different sizes. ReSim also performs inter-image instance discrimination between the two global features (representing the entire positive sample) extracted by the network in order to maintain good performance in classification tasks. MaskCo [23] further introduces the Contrastive Mask Prediction task. It consists in masking one of the local patches (query patch, taken from the first branch), and predicting which augmented view (key views from the second branch) suits the best to fill the masked query patch. Negative key views are introduced by randomly sampling patches from the rest of the dataset, and the contrastive loss is applied to perform the Contrastive Mask Prediction task.

Nevertheless, SCRL, ReSim and MaskCo assume that all overlapping areas are semantically consistent, which may not be the case on dense visual scenes (e.g., if the size of the overlapping area is too large). To avoid this issue, SoCo [24] relies on the selective search algorithm used in Faster R-CNN to extract semantically consistent sub-regions of an image. Furthermore, CAST [25] introduces saliency random cropping. Saliency maps are learned with Grad-CAM supervision, and their goal is to identify foreground objects (and thus semantically consistent regions) within an anchor image. ContrastiveCrop [26] goes further and proposes not only a semantic-aware cropping based on the heatmap analysis during the contrastive training, but also a centre-suppressed sampling (i.e., by limiting center crops) that increases the variance in the crops. Indeed, one issue with random crops is that they may introduce too easy positive pairs. Then, InsLoc [27] and CP^2 [28] introduce background invariance into their crops by copying-pasting foreground images (e.g., crops from ImageNet dataset) on different background images. In their loss, they ensure that the features extracted for the pasted foreground object are similar, regardless of the background.

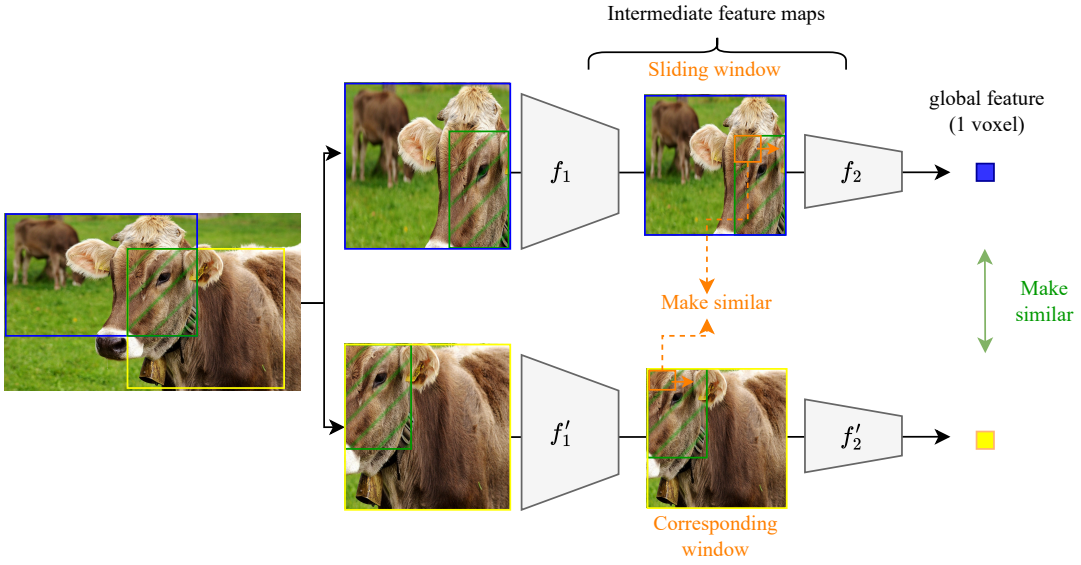

Fig. 2: Example of object-level instance discrimination pipeline. Here, we represented the ReSim framework, which consists in maximizing the similarity between a sliding window in the first branch and its equivalent in the second branch, within an overlapping area.

However, all the methods presented so far rely on intra-image positive pairs, which limits the diversity of information contained in positive pairs. Object-level Representation Learning (ORL) [29] addresses this issue by relying on a three-stage pipeline. First, an instance discrimination method (e.g., BYOL) is trained on an object-centric dataset (e.g., ImageNet) to learn to extract global features. Second, the pre-trained encoder is used to generate local positive pairs. For this purpose, global features are extracted on the target dataset using the pre-trained backbone, and similar images are clustered together using a K-Nearest Neighbors (KNN) algorithm. A selective search algorithm is then used to extract local regions within the similar images, and positive pairs of local patches are matched using the encoder pre-trained in the first step jointly with a KNN clustering. Third, another instance discrimination method is trained using the newly generated local positive pairs. Another alternative is to combine an instance discrimination method based on clustering, such as SwAV, and local augmentations. Leopart [16] builds upon this solution. More specifically, it consists in providing two crops of a foreground object (identified by leveraging ViT attention maps) to an instance discrimination network (e.g., DINO), and then producing patch-level cluster assignments, which are forced to be similar following the online optimization objective of SwAV [30]. Finally, to improve multi-object detection, InsCon [31] ensures multi-instance consistency by taking as a query sample a multi-instance view containing four images, and as positive samples augmentations of each individual image contained in the query sample.

2) Dense loss: The second idea for improving SSL for dense prediction tasks is to apply an instance discrimination loss at "pixel" level (i.e., each voxel of the last feature map), as illustrated on Figure 3. Such a strategy boils down to dividing the image into a grid and taking all (or most of) the patches in the grid into account when computing the instance discrimination loss. The key to this type of method lies in how the positive voxel are matched, i.e. how the features of different views are aligned. In the literature, several alignment strategies have been

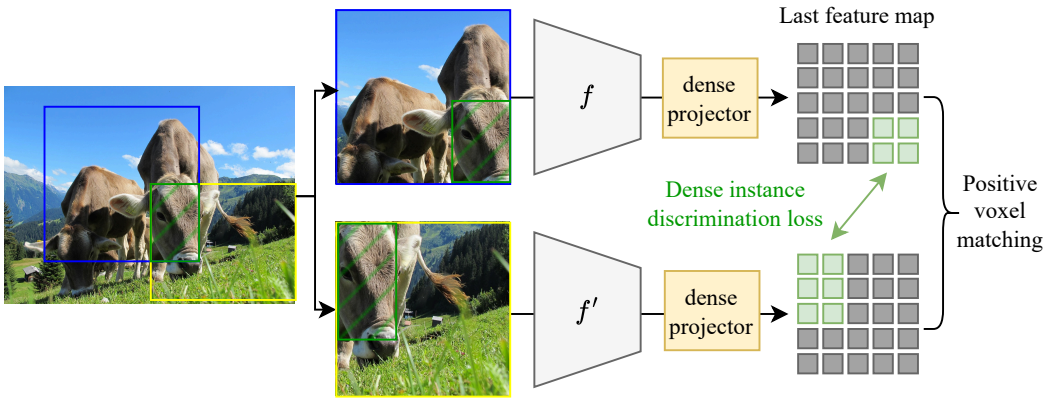

Fig. 3: Dense instance discrimination loss.

proposed:

a) Geometric alignment: VaDeR [32], PixContrast [13], PixPro [13], DUPR [33], InsCon [31], Leopart [16], LC-Loss [34] and CLOVE [35] assume that the geometric transforms between the positive images are known (thanks to the knowledge of the data-augmentation process), and use them to perform spatial alignment. Leopart [16] additionally relies on the attention maps provided by the ViT encoder to focus only on foreground objects in the loss. PixPro further ensures spatial smoothness by propagating the features from similar pixels. CLOVE proposes a similar approach but instead relies on self-attention maps to propagate features.

b) Learned feature matching: It is not always possible to access geometric correspondences, as for example in the case of temporal positive pairs. Therefore, DenseCL [12], Self-EMD [36] and VicRegL [37] align feature voxels that have a minimal distance between their values. An obvious issue with relying solely on feature alignment is that it assumes that the feature extraction is semantically meaningful, which is not the case at the beginning of the training. On the one hand, DenseCL proposes a warm-up before applying this strategy, although they show that random matching (i.e., not semantically consistent matching) also leads to good performance. On the other hand, VicRegL combines learned feature matching with spatial matching.

c) Semantic alignment: To ensure semantic consistency between positive pairs of voxels, DetCon [38] estimates pixel categories (pseudo-labels) through unsupervised segmentation masking (using Felzenszwalb-Huttenlocher algorithm [39]). The authors show empirically that more accurate segmentation masks lead to better fine-tuning performance. In the same line, Odin [14] trains an object *discovery* network together with an instance discrimination pipeline. More specifically, the object *discovery* network relies on K-means clustering to cluster the features in the latent space, assuming that each cluster is more likely to represent an object as the training process progresses. Concurrently, SetSim [15] uses attention maps to estimate both positive pixels location and similar sets of pixels, and then computes the similarity between the sets of pixels.

B. Masked Image Modeling

Conversely to instance discrimination methods whose goal is to estimate some decision borders between image representations, MIM consists in masking a relatively high proportion of an image and reconstructing it (or its features). This brings occlusion invariance to the encoder, as well as locality inductive bias [19]. The underlying hypothesis is that if a network is able to guess or even reconstruct severely corrupted information, then it “understands”

the semantics in the image. Well-known SSL SOTA pipelines such as BEiT [40], Masked AutoEncoders (MAE) [9], iBOT [41] or I-JEPA [42] rely on this principle. They differ mainly in the considered masking strategy and the reconstruction objectives.

1) Masking strategy: In the literature, it has been shown that fine-tuning performance is highly dependent on the masking strategy. We propose to group them by answering the following questions:

a) What shape for the mask?: Authors from MAE [9] and SimMIM [43] evaluate different mask sampling strategies that were previously proposed in the literature, including random, square [44], block-wise [40] and grid masking strategies. The different masking strategies are represented on Figure 4. Both works conclude that the simple random masking strategy is the most efficient, under the condition of considering a high masking ratio. Indeed, such a strategy preserves more hints about the object, especially when considering an object-centric dataset, as opposed to the square and blockwise strategies. Compared to the regular grid masking strategy, random masking brings more difficulties to the network since the object parts are unevenly occluded. Therefore, a semantic understanding of non-occluded patches is necessary to reconstruct some heavily occluded parts. A commonly chosen size for the masked patches is 32 when considering pre-training on images of size 224×224, which has shown to be efficient for many famous computer vision datasets (ImageNet [45], COCO [1], etc.). Note that such patch masking strategies (in terms of size and shape) may not be suitable for some real-world application and data, like in remote sensing or medical domains.The article [46] proposes masks with irregular shapes, which are beneficial for anomaly detection in remote sensing images because the authors simulate the spatial morphology of the anomalies.

b) At which ratio?: Masking a high ratio of patches is also important to make the pretext task difficult enough for the network, forcing it to extract meaningful features. MAE and SimMIM have shown that a ratio of 50% is optimal for random masking. SimMIM further proposes a metric called Average Distance (*AvgDist*) that evaluates the reconstruction difficulty of a given mask sampling strategy. It consists in computing the averaged Euclidean distance between masked pixels and the nearest visible ones. They conclude that masking strategies with an *AvgDist* metric between 10 and 20 have more chance to perform well for fine-tuning. Note that this study has been performed on object-centric datasets (ImageNet, iNaturalist-2018 [47]), as well as on visual scenes (COCO and ADE20K [10]).

c) Which values for the masked areas?: First transformer-based MIM methods propose to replace the masked patches by learnable embeddings [40], [43]. However, MAE showed that encoding masked patches leads to worse results: in addition to a significant impact on the convergence time, it also brings a gap between pre-training and fine-tuning. Indeed, in the fine-tuning task, there are no such corrupted patches. Therefore, the authors of MAE paper propose to encode unmasked patches only, and design a specific decoder that takes as input the masked patches as learnable embeddings.

Another strategy consists in replacing the masked patches by plausible patches. CIM [48] replaces the masking strategy by a more subtle corruption created using a generative network. Such masking strategy seems particularly appropriate for anomaly detection tasks, although the generation of subtle corruptions and their encrustation raise many questions.

d) Where?: Some papers observe that masking patches at random locations can impair the performance of the network [49]. Indeed, if the pre-training dataset contains small objects, they may be totally occluded. The objective of the network will therefore no longer be to reconstruct information but to hallucinate small objects, which poses a problem in terms of learning quality. To avoid this problem, several papers focus on optimizing the masking strategy. The authors of [49] propose a conservative data transform to maintain clues about foreground objects. MST [50], AttMask [51] and AMT [52] rely on self-distillation and use attention maps derived from the teacher network to choose the regions to be masked. MST chooses to mask non-essential regions only, with a low masking ratio ($1/8$), while AttMask shows that masking important features at a moderate ratio ($10 - 50\%$) improves the fine-tuning performance. AMT also relies on attention-driven

a) Original image b) Random c) Square d) Block e) Grid

Fig. 4: Common masking strategies for masked image modelling.

masking, however they use the feature maps derived from MAE or SimMIM last layer attention head (thus they do not rely on siamese architecture for training) after a warm-up phase (40 epochs). Like in [51], AMT makes the most informative parts more likely to be masked although there is still a probability that they remain visible. The authors also show that not using middle attention patches increases the performance, while also reducing the training cost. MILAN [53] also proposes a semantic aware sampling by using attention maps derived from CLIP weights [54] (joint text-image SSL pre-training). However, in contrast to AttMask and AMT, a high probability of remaining unmasked is given to highly informative parts. This is motivated by two elements: i) masking all representative parts of an image leads to very long pre-training, and ii) due to the specific design of their decoder (MAE-like decoder but with frozen representations of the unmasked patches, discussed later), the features extracted by the network need to be informative enough. Indeed, the network should not learn to "hallucinate" objects.

The methods presented so far allow for the decomposition of an image into informative and less informative parts (often foreground/background), and the relationships between those parts (being intra or inter relationships) are learned by the network. What if we further decompose the image by introducing more semantic parts? SemMAE [55] proposes semantic-aware adaptative masking strategy by using some segmentation maps. These segmentation maps are learned in a SSL way by solving a reconstruction task where the targets are patches extracted by a pre-trained ViT (e.g., iBOT), and by adding a diversity constraint on the attention maps. The attention maps obtained are then used for segmenting the image into several semantic parts. The semantic-guided masking of SemMAE then consists of progressively masking 75% of each part (intra-part or local feature learning) at the beginning of the training, to masking 75% of the parts (inter-part relationships) at the end of the training process.

Based on all the masking strategies presented so far, the authors seem to agree on the following conclusions: i) a high masking ratio is recommended to ensure meaningful representation learning, and ii) a carefully designed masking strategy (using either attention or semantic maps) further improves the performance. However, it is not clear which parts should be masked. DPPMask [56] may provide an answer that gets everyone on the same page: keep as much representative and diverse information in unmasked patches, while masking at a high ratio (e.g., 75%). Representative and diverse patches are selected using Determinantal Point Processes (DPP), which aim at reducing the semantic change of an image after masking (miss-alignment problem). It consists in computing the distance (using a Gaussian kernel, which depends on the Euclidean distance between the intensity values of the patch pairs) between each patch and selecting those that are dissimilar from a selected subset. Due to the computational complexity resulting from the exact DPP formulation (matrix decomposition), DPPMask proposes a greedy approximation of DPP. DPPMask shows significant improvements over AttMask and SemMAE masking strategies for both MAE and iBOT.

2) Reconstruction targets: Although masking strategy is very important to improve the performance, there are also many discussions about the

choice of the reconstruction targets. First MIM methods [57], [9], [43] attempt to reconstruct raw pixels, and apply the Mean Squared Error (MSE) loss as the reconstruction objective. An important limitation with such a reconstruction objective is that all reconstructed pixels have the same weight in the loss, although some reconstruction errors may be irrelevant for meaningful feature extraction.

Therefore, some methods propose to adapt the reconstruction target to the downstream objectives. For example, to force the network to focus on shapes rather than texture and rich details, PixMIM [49] filters the high frequencies in the target objective (and thus the network focuses on low frequencies). Ge^2-AE [58] and A^{2}MIM [59] apply the reconstruction loss in both spatial and frequency domains to learn global features. In the same spirit, MaskFeat [60] uses the Histogram of Oriented Gradients (HOG) as a reconstruction target, and justify this choice by the fact that HOG provides local shapes and appearances while being invariant to photometric changes. In the same line, SSM [61] applies different reconstruction losses which introduce some global criteria that do not suppose independence between neighboring pixels, such as Gradient Magnitude Similarity (GMS) and Structured Similarity Index Measure (SSIM).

However, all these methods rely on computationally expensive architectures in order to reconstruct the full-resolution (or almost) image, with a decoder that will not be used for the final task. To address this issue, some authors propose to reconstruct some features instead of full-resolution images. In this case, the challenge consists in defining relevant target features. Among the ideas proposed and tested, the literature has retained

- **Features from a pre-trained network –** Several methods, such as BEiT [40], MaskDistill [62], MILAN [53] and MaskAlign [63], rely on distillation from strong unsupervised pre-trained encoders, such as CLIP. However, such a strategy may not be optimal on datasets that present a domain gap (e.g., satellite data) with, for example, CLIP pre-training data.
- **Features obtained via self-distillation –** Another way to obtain target features is by relying on self-distillation methods and asymmetric siamese networks. Such a strategy is adopted by SOTA methods like SplitMask [64], iBOT and I-JEPA [42]. I-JEPA differs from iBOT by the fact that it asks the network to reconstruct not the full masked areas, but only parts of the masked image given a context. Nonetheless, the target features obtained using pre-trained weights seem to lead to better representation learning. [65] claims that it is not necessary to carefully choose the target (HOG, MaskFeat or features obtained with MAE/SimMIM etc.) as long as a multi-stage distillation pipeline is used, which leads to dBOT method. However, even with dBOT framework, CLIP pre-trained teacher still leads to better performance than a randomly initialized teacher.

In the literature, many questions have been raised about the design of the decoder in the SSL pre-training phase. Some authors argue that it is better to use a simple decoder to maximize transfer learning performance [43], while others have observed that a deep and narrow decoder works best [9]. This is one of the open questions in the field of image reconstruction. Indeed, how can we ensure that it is the encoder and not the decoder that learns to extract highly representative information from an image and to disentangle causal factors? MILAN [53] proposes to circumvent this issue by designing a specific decoder that clearly separates the functional roles of the encoder and the decoder. To this end, the authors introduce a prompting decoder that takes as input frozen representations of encoded unmasked patches. The latter are therefore used as fixed prompts. However, the ablation study shows that the SOTA performance achieved with MILAN is mainly due to the use of CLIP targets and not to the design of the prompting decoder.

3) Adaptation to convolutional networks: Most papers tackling MIM rely on the use of ViT encoders. CNN-based encoders are still widely used in many real-world applications and have some advantages, such as faster inference times on small inputs, and a hierarchical architecture that benefits object detection. However, they seem to be less efficient than ViT encoders when combined with MIM methods. This may be due to their poor ability to estimate

large-scale relationships between image patches. Also, unlike ViT architectures that analyze each patch independently, CNN-based encoders perform convolutions by sliding a window, and thus the receptive field of the convolution can overlap with both masked and unmasked areas. This leads to several issues such as masked pattern vanishing or the disturbance of the distribution of pixel values, as explained in [66]. A^2MIM [59] attempts to solve this issue by replacing the 0-padding with a padding the mean value of the unmasked pixels. ConvMAE [67], MixMAE [68] and SparK [66] introduce the use of partial or sparse convolutions. Specifically, the authors of the SparK [66] paper show that MAE pre-training with a CNN-based encoder can outperform ViT-based MAE pre-training when using sparse convolution and a modern CNN-based encoder, namely ConvX-B [69]. Furthermore, [68] efficiently encodes two images as a single image by replacing the masked patches of the first image (image 1) with the unmasked ones of the second image (image 2). To adapt this strategy to ConvNets, they introduce unmixed convolutions, which consists in unmixing the image into image 1 and image 2, and then applying partial convolution.

III. Benchmark on the COCO dataset

Now that we have introduced the key SSL strategies for enhancing object detection, we propose to evaluate a selection of them on two benchmark datasets. The considered SSL methods are summarized in Table II, and the object detection framework corresponds to the well-established and widely-used dataset from the literature, namely the COCO dataset [1]. Various sizes of objects are covered, including 41% of small objects (i.e., objects having an area lower than 32^2 pixels). In the literature, although a large number of SSL papers evaluate their methods on the COCO dataset, the fine-tuning set-ups or the evaluation conditions may differ from one paper to another. To ensure a fair comparison, we propose to fine-tune the studied SSL methods ourselves, using training parameters from recent papers that have proven their efficiency.

Method	Category	Backbone	#params
DINO [5]	Inst. Discr. (global)	R50	23M
		ViT/S-16	21M
		ViT/B-16	83M
ReSim [11]	Inst. Discr. (local)	R50	23M
Leopart [16]	Inst. Discr. (local)	ViT/S-16	21M
SparK [66]	MIM	R50	23M
		R200	65M
MAE [9]	MIM	ViT/S-16	21M
		ViT/B-16	83M

TABLE II: Compared pre-training methods, along with their SSL category, considered backbones and number of parameters in each backbone. R50 stands for ResNet-50 backbone, R200 for ResNet-200, ViT/S-16 for Vision Transformer (ViT) Small version with a patch size of 16, and ViT/B-16 for ViT Base version with a patch size of 16. "Inst. Discr." stands for instance discrimination methods and "MIM" for masked image modeling.

A. Experimental set-up

We consider a Mask R-CNN [70] with ResNet-50 (R50), ResNet-200 (R200), ViT/B-16 or ViT/S-16 encoders as our detectors. For the encoder, the pre-trained weights of each SSL method are taken from the Github repository published by the authors of the original papers. The fine-tuning parameters for the ResNet-based encoders (namely R50 and R200) are chosen following SparK's paper [66] recommendations. More specifically, we train the detector using AdamW optimizer [71] and the $3\times$ schedule (i.e., we trained the network for 3×12 epochs). For the learning rate, since we can only load 36 images on our GPUs, we use the linear scaling rule introduced in [72] to choose an appropriate learning rate. We consider the "Step LR" scheduler, and multiply the learning rate by 0.2 at epochs 3×9 and 3×11. For ViT-based fine-tuning, we follow the training set-up proposed in [73] and scale the learning rate according to our GPU resources (four Nvidia A100 GPUs) based on the linear scaling rule. We fine-tune the neural networks for 50 epochs using

AdamW optimizer and CosineLR scheduler.

We use "COCO 2017 val" subset as our test set and evaluate the box location accuracy of each method using the conventional mean average precision metric $\text{mAP}^{box}_{@0.5:0.95}$ (i.e., the area under the precision-recall curve, averaged over all the object classes and over 10 IoU threshold from 0.5 to 0.95). In order to focus on the detection performance, we will also provide the metrics for box location regardless of the errors made on the classification ($\text{AP}^{box}_{@0.5:0.95}$). We will also focus on small object detection performance by providing these metrics for objects that have a spatial extent less than 32×32 pixels ($\text{AP}^{box,S}_{@0.5:0.95}$, $\text{AP}^{box,S}_{@0.3}$). Since a small deviation in the box localization for small objects drastically reduces the IoU between the predicted box and the ground-truth, we introduce more tolerance regarding the localization errors by lowering the IoU threshold to 30% ($\text{AP}^{box}_{@0.3}$, $\text{AP}^{box,S}_{@0.3}$).

B. Reproducibility

First of all, we would like to make a few comments about the reproducibility of the results presented in the original papers.

For the methods trained with a ResNet-50 encoder, the results we have obtained are slightly better than those presented in the original papers. This difference can be explained by the choice of a longer schedule, along with a different optimizer, namely AdamW optimizer instead of the classical SGD optimizer.

For ViT-based fine-tuning, the results we have obtained are worse than those reported in the original papers. For example, [9] achieves a $\text{mAP}^{box}_{@0.5:0.95}$ of 50.3% on the COCO dataset using MAE pre-trained weight, while we can only achieve a $\text{mAP}^{box}_{@0.5:0.95}$ of 47.8% (-2.5%). This can be partly explained by the fact that we considered a shorter fine-tuning schedule (only 50 epochs instead of 100 epochs in [9]). Moreover, since we did not have access to the same amount of GPU resources as the original papers, we were forced to drastically reduce the size of our batches. Despite adapting the learning rate accordingly, it is likely that the linear scaling rule [72] does not directly apply, meaning that our training parameters are not optimal. Due to the excessive computation time, the search for optimal training parameters has been set aside, and it must therefore be assumed that there is a slight difference in the results, of about 2% or 3%.

C. Results

Table III presents the results obtained on COCO-val 2017 dataset. Our observations are the following:

a) The encoder architecture matters more than the SSL strategy: According to Table III, large networks, especially those based on ViT/B-16 backbone, lead to the best results. For example, the $\text{mAP}^{box}_{@0.5:0.95}$ is increased by 2.6% when considering a ResNet-200 encoder instead of a ResNet-50 encoder for SparK, and increased by 1% when considering a ViT/B encoder instead of a ViT/S for DINO. Note that the performance gap is narrower for ViT encoders than with CNN. Moreover, ViT backbones perform significantly better than ResNet backbones. However, the performance gap is reduced if classification errors are ignored, especially when it comes to small objects. Indeed, Table III shows that SparK initialization on a ResNet-200 encoder leads to an $\text{AP}^{box,S}_{@0.5:0.95}$ that is 1.8% better than MAE initialization on a ViT/B-16. We deduce that ResNet encoders are likely to be more prone to classification errors than ViT encoders.

b) Introducing locality in the SSL pre-training is important for ResNet-based encoders: Let us now take a closer look at the performance obtained by each SSL strategy. Concerning ResNet-50 backbone, it is clear that ReSim outperforms the other pre-training strategies. SparK (MIM method) leads to competitive performance, while DINO seems to be the worst SSL training strategy for this task. The results seem to be consistent with our intuition: in contrast to global instance discrimination, both local instance discrimination and MIM methods force the neural networks to model local interactions within the image, which may benefit object detection. When looking at the detection performance only (i.e., no classification), we notice that, for ResNet-50 backbones, ReSim and SparK lead to very close results even on small objects, although ReSim is slightly better than SparK when lowering the IoU

Backbone		With Class.	No Class.		Small objects, no Class.	
		$mAP^{box}_{@0.5:0.95}$	$AP^{box}_{@0.5:0.95}$	$AP^{box}_{@0.3}$	$AP^{box,S}_{@0.5:0.95}$	$AP^{box,S}_{@0.3}$
Small networks (21-23 M #params.)						
Instance discrimination methods						
DINO	R50	42.8	46.9	77.0	31.9	64.0
ReSim	R50	44.3	48.6	78.4	**33.3**	65.7
DINO	ViT/S-16	<u>46.3</u>	<u>48.8</u>	<u>79.9</u>	32.2	<u>66.9</u>
Leopart	ViT/S-16	**46.5**	**49.0**	**80.1**	32.4	**67.1**
MIM methods						
SparK	R50	44.1	48.6	78.0	**33.3**	64.9
Large networks ($\geq$ 65 M #params.)						
Instance discrimination methods						
DINO	ViT/B-16	<u>47.3</u>	49.1	<u>80.2</u>	32.7	**68.1**
MIM methods						
SparK	R200	46.7	**50.5**	79.4	**35.2**	67.5
MAE	ViT/B-16	**47.8**	<u>50.3</u>	**80.7**	<u>33.4</u>	<u>67.6</u>

TABLE III: Benchmark on the COCO dataset with ("With Class.") or without classification labels ("No Class.", i.e., detection only). For each network size (small or large), the best results are in bold and the second best results are underlined.

threshold. The performance gap with DINO remains very large, especially for small objects.

c) ViT encoders are less sensitive to the pre-training strategy: For ViT encoders, the difference in performance between the SSL strategies is very thin: although local methods (MIM or local instance discrimination methods) seem to perform slightly better in terms of $AP^{box}_{@0.5:0.95}$, introducing more tolerance towards localization errors shows that DINO with ViT/S-16 or ViT/B-16 encoder is also very competitive on small object detection. Furthermore, DINO with ViT/B-16 encoder leads to the best $AP^{box,S}_{@0.3}$ score. This suggests that, in an ideal and data-sufficient case, the ViT backbones are less sensitive to the pre-training strategy compared to the ResNet encoders.

d) ViT encoders are more prone to localization errors on small objects: Still referring to Table III, the $AP^{box,S}_{@0.5:0.95}$ column shows that ResNet-based encoders lead to the best performance on small objects (e.g., $+0.9\%$ in $AP^{box,S}_{@0.5:0.95}$ when comparing Leopart and ReSim), meaning that these architectures are better suited to small object detection. Nevertheless, the introduction of greater tolerance to localization errors reveals that ViT encoders are still capable of detecting small objects, albeit with a slightly worse localization accuracy.

IV. What about domain-specific tasks?

In this section, we challenge the previously studied SSL pre-training strategies in a real-world scenario, namely small vehicle detection from remote sensing data. For this purpose, we consider the VEDAI dataset [2], which is composed of 1200 RGB and IR satellite scenes containing small vehicles. This allows us to study the cross-domain transfer ability of the considered pre-training strategies, from RGB to IR domain. We will try to answer the following questions: 1) does SSL pre-training benefit real-world small object detection? 2) is it better to perform SSL pre-training on a dataset whose statistics are close to those of the target data? (e.g., infrared dataset, remote sensing data), 3) which SSL strategy is best for pre-training on an uncleaned dataset (i.e., with high temporal redundancy, low diversity, etc.)?, and 4) can SSL benefit few-shot training?

A. Experimental set-up

We fine-tune a Faster R-CNN on the RGB version of the VEDAI dataset with various encoders

Backbone		VEDAI RGB		VEDAI IR	
		AP	F1	AP	F1
Small networks (21-23 M #params.)					
Scratch	R50	61.8	62.5	61.3	60.9
Scratch	ViT/S-16	79.4	72.8	74.8	71.3
Instance discrimination methods					
DINO	R50	86.1	82.0	84.0	79.0
ReSim	R50	87.7	84.4	<u>85.1</u>	<u>81.6</u>
DINO	ViT/S-16	89.7	81.8	84.4	78.1
Leopart	ViT/S-16	<u>91.0</u>	<u>84.5</u>	84.3	78.0
MIM methods					
SparK	R50	86.4	83.2	81.1	78.4
MAE	ViT/S-16	**91.8**	**86.1**	**88.4**	**83.7**
Large networks ($\geq$ 65 M #params.)					
Scratch	ViT/B-16	66.7	63.2	58.5	57.3
Instance discrimination methods					
DINO	ViT/B-16	**94.9**	**89.6**	**90.7**	<u>85.6</u>
MIM methods					
MAE	ViT/B-16	<u>94.1</u>	<u>88.5</u>	**92.1**	**86.0**

TABLE IV: Benchmark of different pre-training methods on the VEDAI RGB and IR datasets. For each network size (small or large), the best results are in bold and the second best results are underlined.

initialized with different pre-training strategies (SSL or supervised on ImageNet). The training parameters are those used in Section III-A, except that we considered a CosineLR scheduler for ResNet-based architectures since it leads to better performance. We split the VEDAI dataset into training, validation and test sets using a ratio of $60:20:20$, and consider the AP (with an IoU threshold of 5%) and F1 score metrics for evaluation. Since ViT/S-16 weights pre-trained using MAE strategy are not available in the literature, we decided to perform MAE pre-training on ImageNet dataset ourselves. We used the same training parameters as in the original paper and trained the encoder for 400 epochs.

B. Results obtained on the VEDAI RGB dataset

According to Table IV, on the RGB version of VEDAI dataset, there is a large gap between the performance obtained using a ResNet-50 and a ViT encoder. In particular, the use of large ViT encoders leads to impressive performance on this dataset. For example, a ViT/S-16 encoder can achieve an AP of almost 92%, while ResNet encoders merely reach an AP of 87.7%. Let us now dive into the performance achieved by the different SSL strategies. For ResNet-50 backbones, ReSim pre-training performs significantly better than DINO and SparK pre-training strategies. For ViT backbones, it is difficult to draw conclusions: MAE seems to benefit the most for small encoder pre-training, while DINO performs slightly better than MAE with a larger encoder. It seems that, for ViT encoders, the fine-tuning performance on the final task is less dependent on the ViT initialization, which is in line with what was observed on the COCO dataset. In the end, it seems that the choice of a good encoder, especially those based on ViT blocks, is more important for the performance of the downstream task than the choice of a good pre-training strategy. But what if we consider a downstream task dataset whose image statistics are very different from those of ImageNet?

C. Transferring the knowledge learned on RGB data to IR domain

We now evaluate the ability of the different pre-training strategies to transfer to other spectral domains using IR imagery as a target example. For this purpose, we consider the IR images of VEDAI dataset and coined this subset of data as VEDAI IR. We fine-tune a Faster R-CNN with different pre-trained encoders in the same way as previously. Note that these encoders have been pre-trained on RGB images (ImageNet dataset). The last two columns of Table IV show the results obtained on VEDAI IR dataset. We first notice that there is a large drop in performance for ViT-based instance discrimination pre-training strategies, and they perform even worse than the ResNet-based pre-trainings (for equivalent network size). Indeed, DINO and Leopart pre-training strategies with ViT/S-16 perform about 5% worse in $\text{AP}^{box}_{@0.05}$ when applied to VEDAI IR dataset, while MAE leads to a decrease of only 2%. The performance gap is less pronounced when it comes to larger networks, and MAE leads to the best performance.

For ResNet backbones, ReSim seems to be significantly more robust than any other pre-training strategy, while SparK suffers from a large drop in performance. According to these observations, the fine-tuning performance of SSL pre-trained weights varies greatly depending on the encoder architecture considered: MIM methods combined with ViT encoders seem to generalize better to datasets that statistically differ from the ImageNet dataset, whereas in the case of ResNets, it is the instance discrimination methods that perform best. This may be explained by the fact that MIM methods are very sensitive to the image statistics, due to their strong bias towards local details (e.g., textures), and may therefore show a decrease in performance when applied to a different dataset. However, since ViT encoders are better at modeling large-scale dependencies (i.e., they have a bias towards shapes), the combination of ViT encoders and MIM methods compensates for the weakness observed for the latter. Thus the following question arises: can we improve the performance by pre-training on a dataset that has close characteristics to the downstream task dataset? To answer this question, we perform some SSL pre-training on an infrared dataset, that however is uncleaned (i.e., without removal of redundant images). Results are commented in the next paragraph. This will also allow us to assess the degree of generalization ability of SSL pre-training to other pre-training databases.

D. Pre-training on an uncleaned infrared dataset

To be able to perform SSL pre-training on an infrared dataset, we collected a large number of infrared images from several publicly available infrared datasets. Table V summarizes the different infrared dataset sources that we merged together in order to obtain a large infrared dataset, and we coined the final dataset as **SSL-IR** dataset. The datasets we used to obtain SSL-IR have very different characteristics: they contain different scenes (urban, sky, forest...) captured from various camera viewpoints (drone, car), and with different infrared sensors (thermal infrared, near infrared, etc.). However, most images are extracted from video sequences, and thus the obtained dataset suffers from low image diversity. We obtain a total of approximately 720k infrared images, which represents about 60% of ImageNet-1k dataset.

We pre-trained ReSim (R50), SparK (R50), Leopart (ViT/S-16) and MAE (ViT/S-16) on the SSL-IR dataset using the pre-training parameters suggested for each method in the original papers. We then fine-tuned a Faster R-CNN on VEDAI IR under the same conditions as before. The results are shown in Table VI. According to this table, ReSim suffers from a huge drop in performance (more than 8% in both AP and F1 score), while the decrease in performance is limited for SparK and Leopart. Moreover, MAE is particularly robust to training on SSL-IR dataset, since the performance is almost equivalent to the pre-training on ImageNet. Overall, for both ResNet and ViT encoders, MIM-based SSL pre-training is more robust to pre-training on a smaller and less clean dataset than its instance discrimination counterparts. At first sight, the results of the pre-training on SSL-IR are rather disappointing compared to the RGB weights available in the literature. However, it should be remembered that the IR dataset we considered is not cleaned and is even much smaller than ImageNet. Furthermore, by choosing the right SSL strategy and encoder, we can obtain results that are very similar to those given by the weights in the literature. This is encouraging, especially in cases where it is absolutely necessary to pre-train SSL on custom datasets (for example, if there is a large domain gap, or if the encoder architecture needs to be significantly changed).

E. Frugal setting

Finally, we evaluate the different SSL strategies in challenging fine-tuning conditions, namely few-shot setting. For this purpose, we consider fine-tuning on 25 or 50 images from the VEDAI RGB dataset. The results are presented in Table VII. In general, we can see that there is a real contribution of using SSL pre-trained weights in few-shot setting, although the benefits are more or less obvious depending on the SSL strategies or architectures used. Firstly, we can see that ViT/S encoders achieve significantly inferior performance compared to ResNet-50, even when relying on SSL pre-trained weights. Secondly, the

Source dataset	Type of data	Nature of data	# images
LSOTB-TIR [74]	drone, car, fixed cameras, urban sky natural scenes, thermal infrared object tracking	video	524k
IRDST [75]	real and simulated data, drone, sky and urban scene, target detection	video	143k
FLIR [76]	car, urban scenes, autonomous driving	video	35k
MFIRST [77]	drone, sky and urban scenes, simulated and real small target detection	single-frame images	10k
ASL-TID [78]	drone, urban scenes, pedestrian detection	video	4k
HIT-UAV [79]	drone, urban scenes, pedestrian detection	video	3k
IRSTD-1k [80]	drone, sky, natural and urban scenes, small target detection	single-frame images	1k
SSL-IR			**720k**

TABLE V: SSL-IR dataset: data sources and specifications.

	Backbone	$AP^{box}_{@0.05}$	F1
Scratch	R50	61.3	60.9
Instance discrimination methods			
ReSim-IR	R50	$76.6^{(-8.5)}$	$72.9^{(-8.7)}$
Leopart-IR	ViT/S-16	$81.6^{(-2.7)}$	$76.7^{(-1.3)}$
MIM methods			
SparK-IR	R50	$77.4^{(-3.7)}$	$75.0^{(-3.4)}$
MAE-IR	ViT/S-16	$\mathbf{88.5}^{(-0.1)}$	$\mathbf{82.8}^{(-0.9)}$

TABLE VI: Benchmark on VEDAI IR with SSL methods pre-trained on SSL-IR dataset. The best results are in bold, and the performance gaps with the respective SSL strategies pre-trained on ImageNet are indicated in the superscript.

Backbone		25-shots		50-shots	
		AP	F1	AP	F1
Scratch	R50	30.1	22.2	33.9	25.9
Scratch	ViT/S-16	14.8	6.8	24.2	13.9
Instance discrimination methods					
DINO	R50	<u>38.2</u>	<u>33.4</u>	<u>53.4</u>	<u>53.3</u>
ReSim	R50	**50.4**	**52.0**	**57.5**	**58.2**
DINO	ViT/S-16	20.1	8.9	30.9	21.5
Leopart	ViT/S-16	17.7	9.5	30.3	21.9
MIM methods					
SparK	R50	34.4	29.2	48.8	44.7
MAE	ViT/S-16	33.9	29.2	42.4	37.2

TABLE VII: Results obtained on VEDAI RGB in 25 and 50-shot settings. The best results are in bold and the second best results are underlined.

choice of the SSL pre-training strategy depends on the encoder. For ResNet-50, instance discrimination methods, in particular ReSim, significantly benefits 25 and 35-shot trainings. This is evidenced by an improvement of over 20% in terms of AP and F1 score when compared to a network that has been trained from scratch. SparK exhibits only marginal improvement over randomly initialized weights, especially in the 25-shot setting. Regarding the results obtained with a ViT/S encoder, a notable improvement is observed when ViT is combined with the MIM method (specifically MAE), although the performance remains inferior to that observed with ResNet-50.

V. Conclusion

In this paper, we presented a survey of SSL strategies oriented towards local feature extraction, which appear better suited to object detection tasks. We performed a benchmark using two distinct datasets: 1) the COCO dataset, which represents an ideal scenario for object detection with a large amount of diverse data, and 2) the VEDAI dataset, a real-world, domain-specific case with IR images that deals with much smaller objects and more complex, diverse backgrounds, making it quite different from the ImageNet dataset used by the authors to pre-train their SSL methods. These benchmarks allowed us to draw important conclusions to guide future users

in choosing appropriate pre-training strategies based on their specific use cases. The key takeaways are:

- **Importance of the encoder choice:** The selection of the encoder is more critical than the choice of the pre-training strategy. ViT encoders generally outperform ResNets when sufficient fine-tuning data is available and when dealing with large objects. In this case, ViTs are less sensitive to the pre-training strategy. However, they tend to perform poorly in frugal settings, and should be combined with MIM methods in such cases.
- **ResNets are more sensitive to the SSL pre-training:** ResNets perform better when combined with local instance discrimination methods or MIM. However, MIM pre-training leads to poor performance in a frugal setting.
- **Domain shift:** We observed that pre-training on in-domain images does not necessarily improve performance and may even degrade it in the considered case, namely from RGB to IR. This might be because IR images are still relatively close to RGB, which explains why weights pre-trained on RGB data can generalize well to IR data. However, conclusions might differ with more significant domain shifts (e.g., astronomy or medical images), and SSL pre-training on a custom (and maybe uncleaned dataset) may be necessary. In this case, MIM methods should be prioritized for both ViT and ResNet networks. Note however that if the downstream tasks deals with frugal dataset, combining MIM and instance discrimination as in CMAE [81] or Siamese image modelling [82] could yield better results for ResNets.

Future work should focus on providing more theoretical explanations for the differences in behavior of pre-trained SSL strategies depending on the encoder. For example, we hypothesize that because ViTs model long-range dependencies, they are well-complemented by local SSL methods such as MIM. This hypothesis needs further investigation. Additionally, exploring other application areas, such as anomaly detection, could help expand and complete this benchmark.

Acknowledgments – This project was provided with computing HPC and storage resources by GENCI at IDRIS thanks to the grant 2023-AD011014896 on the supercomputer Jean Zay's V100 and A100 partitions. It was also performed using computational resources from the "Mésocentre" computing center of Université Paris-Saclay, CentraleSupélec and École Normale Supérieure Paris-Saclay supported by CNRS and Région Île-de-France.

REFERENCES

[1] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. Springer, 2014, pp. 740–755.

[2] S. Razakarivony and F. Jurie, "Vehicle detection in aerial imagery: A small target detection benchmark," Journal of Visual Communication and Image Representation, vol. 34, pp. 187–203, 2016.

[3] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 9729–9738.

[4] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar et al., "Bootstrap your own latent-a new approach to self-supervised learning," Advances in neural information processing systems, vol. 33, pp. 21 271–21 284, 2020.

[5] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 9650–9660.

[6] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A Survey on Contrastive Self-supervised Learning," arXiv:2011.00362, Feb. 2021. [Online]. Available: http://arxiv.org/abs/2011.00362

[7] U. Ozbulak, H. J. Lee, B. Boga, E. T. Anzaku, H. Park, A. Van Messem, W. De Neve, and J. Vankerschaver, "Know your self-supervised learning: A survey on image-based generative and discriminative training," arXiv preprint arXiv:2305.13689, 2023.

[8] J. Gui, T. Chen, J. Zhang, Q. Cao, Z. Sun, H. Luo, and D. Tao, "A survey on self-supervised learning: Algorithms, applications, and future trends," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024.

[9] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 16 000–16 009.

[10] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ade20k dataset," International Journal of Computer Vision, vol. 127, pp. 302–321, 2019.

[11] T. Xiao, C. J. Reed, X. Wang, K. Keutzer, and T. Darrell, "Region similarity representation learning," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10 539–10 548.

[12] X. Wang, R. Zhang, C. Shen, T. Kong, and L. Li, "Dense contrastive learning for self-supervised visual pre-training," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3024–3033.

[13] Z. Xie, Y. Lin, Z. Zhang, Y. Cao, S. Lin, and H. Hu, "Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 16 684–16 693.

[14] O. J. Hénaff, S. Koppula, E. Shelhamer, D. Zoran, A. Jaegle, A. Zisserman, J. Carreira, and R. Arandjelović, "Object discovery and representation networks," in European Conference on Computer Vision. Springer, 2022, pp. 123–143.

[15] Z. Wang, Q. Li, G. Zhang, P. Wan, W. Zheng, N. Wang, M. Gong, and T. Liu, "Exploring set similarity for dense self-supervised representation learning," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16 590–16 599.

[16] A. Ziegler and Y. M. Asano, "Self-supervised learning of object parts for semantic segmentation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 14 502–14 511.

[17] Y. Li, S. Xie, X. Chen, P. Dollar, K. He, and R. Girshick, "Benchmarking detection transfer learning with vision transformers," arXiv preprint arXiv:2111.11429, 2021.

[18] N. Park, W. Kim, B. Heo, T. Kim, and S. Yun, "What do self-supervised vision transformers learn?" in The Eleventh International Conference on Learning Representations, 2022.

[19] Z. Xie, Z. Geng, J. Hu, Z. Zhang, H. Hu, and Y. Cao, "Revealing the dark secrets of masked image modeling," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 14 475–14 485.

[20] J. Gao, S. Lin, S. Wang, Y. Kou, Z. Li, L. Li, C. Zhang, X. Zhang, Y. Wang, and W. Hu, "Observation, analysis, and solution: Exploring strong lightweight vision transformers via masked image modeling pre-training," arXiv preprint arXiv:2404.12210, 2024.

[21] G. Huang, I. Laradji, D. Vazquez, S. Lacoste-Julien, and P. Rodriguez, "A survey of self-supervised and few-shot object detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 4, pp. 4071–4089, 2022.

[22] B. Roh, W. Shin, I. Kim, and S. Kim, "Spatially consistent representation learning," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 1144–1153.

[23] Y. Zhao, G. Wang, C. Luo, W. Zeng, and Z.-J. Zha, "Self-supervised visual representations learning by contrastive mask prediction," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10 160–10 169.

[24] F. Wei, Y. Gao, Z. Wu, H. Hu, and S. Lin, "Aligning pretraining for detection via object-level contrastive learning," Advances in Neural Information Processing Systems, vol. 34, pp. 22 682–22 694, 2021.

[25] R. R. Selvaraju, K. Desai, J. Johnson, and N. Naik, "Casting your model: Learning to localize improves self-supervised representations," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 11 058–11 067.

[26] X. Peng, K. Wang, Z. Zhu, M. Wang, and Y. You, "Crafting better contrastive views for siamese representation learning," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16 031–16 040.

[27] C. Yang, Z. Wu, B. Zhou, and S. Lin, "Instance localization for self-supervised detection pretraining," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3987–3996.

[28] F. Wang, H. Wang, C. Wei, A. Yuille, and W. Shen, "Cp 2: Copy-paste contrastive pretraining for semantic segmentation," in European Conference on Computer Vision. Springer, 2022, pp. 499–515.

[29] J. Xie, X. Zhan, Z. Liu, Y. S. Ong, and C. C. Loy, "Unsupervised object-level representation learning from scene images," Advances in Neural Information Processing Systems, vol. 34, pp. 28 864–28 876, 2021.

[30] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," Advances in neural information processing systems, vol. 33, pp. 9912–9924, 2020.

[31] J. Yang, K. Zhang, Z. Cui, J. Su, J. Luo, and X. Wei, "Inscon: Instance consistency feature representation via self-supervised learning," arXiv preprint arXiv:2203.07688, 2022.

[32] P. O. O Pinheiro, A. Almahairi, R. Benmalek, F. Golemo, and A. C. Courville, "Unsupervised learning of dense visual representations," Advances in Neural Information Processing Systems, vol. 33, pp. 4489–4500, 2020.

[33] J. Ding, E. Xie, H. Xu, C. Jiang, Z. Li, P. Luo, and G.-S. Xia, "Deeply unsupervised patch re-identification for pre-training object detectors," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022.

[34] A. Islam, B. Lundell, H. Sawhney, S. N. Sinha, P. Morales, and R. J. Radke, "Self-supervised learning with local contrastive loss for detection and semantic segmentation," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2023, pp. 5624–5633.

[35] T. Silva, H. Pedrini, and A. Ramírez, "Self-supervised learning of contextualized local visual embeddings," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 177–186.

[36] S. Liu, Z. Li, and J. Sun, "Self-emd: Self-supervised object detection without imagenet," arXiv preprint arXiv:2011.13677, 2020.

[37] A. Bardes, J. Ponce, and Y. LeCun, "Vicregl: Self-supervised learning of local visual features," Advances in Neural Information Processing Systems, vol. 35, pp. 8799–8810, 2022.

[38] O. J. Hénaff, S. Koppula, J.-B. Alayrac, A. Van den Oord, O. Vinyals, and J. Carreira, "Efficient visual pretraining with contrastive detection," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10 086–10 096.

[39] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient

graph-based image segmentation," International journal of computer vision, vol. 59, pp. 167–181, 2004.

[40] H. Bao, L. Dong, S. Piao, and F. Wei, "Beit: Bert pre-training of image transformers," in International Conference on Learning Representations, 2021.

[41] J. Zhou, C. Wei, H. Wang, W. Shen, C. Xie, A. Yuille, and T. Kong, "Image bert pre-training with online tokenizer," in International Conference on Learning Representations, 2021.

[42] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, Y. LeCun, and N. Ballas, "Self-supervised learning from images with a joint-embedding predictive architecture," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 15 619–15 629.

[43] Z. Xie, Z. Zhang, Y. Cao, Y. Lin, J. Bao, Z. Yao, Q. Dai, and H. Hu, "Simmim: A simple framework for masked image modeling," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 9653–9663.

[44] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2536–2544.

[45] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in 2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009, pp. 248–255.

[46] Z. Li, Y. Wang, C. Xiao, Q. Ling, Z. Lin, and W. An, "You only train once: Learning a general anomaly enhancement network with random masks for hyperspectral anomaly detection," IEEE Transactions on Geoscience and Remote Sensing, vol. 61, pp. 1–18, 2023.

[47] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie, "The inaturalist species classification and detection dataset," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8769–8778.

[48] Y. Fang, L. Dong, H. Bao, X. Wang, and F. Wei, "Corrupted image modeling for self-supervised visual pre-training," in The Eleventh International Conference on Learning Representations, 2022.

[49] Y. Liu, S. Zhang, J. Chen, K. Chen, and D. Lin, "Pixmim: Rethinking pixel reconstruction in masked image modeling," Transactions on Machine Learning Research, 2024.

[50] Z. Li, Z. Chen, F. Yang, W. Li, Y. Zhu, C. Zhao, R. Deng, L. Wu, R. Zhao, M. Tang et al., "Mst: Masked self-supervised transformer for visual representation," Advances in Neural Information Processing Systems, vol. 34, pp. 13 165–13 176, 2021.

[51] I. Kakogeorgiou, S. Gidaris, B. Psomas, Y. Avrithis, A. Bursuc, K. Karantzalos, and N. Komodakis, "What to hide from your students: Attention-guided masked image modeling," in European Conference on Computer Vision. Springer, 2022, pp. 300–318.

[52] Z. Liu, J. Gui, and H. Luo, "Good helper is around you: Attention-driven masked image modeling," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, no. 2, 2023, pp. 1799–1807.

[53] Z. Hou, F. Sun, Y.-K. Chen, Y. Xie, and S.-Y. Kung, "Milan: Masked image pretraining on language assisted representation," arXiv preprint arXiv:2208.06049, 2022.

[54] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark et al., "Learning transferable visual models from natural language supervision," in International conference on machine learning. PMLR, 2021, pp. 8748–8763.

[55] G. Li, H. Zheng, D. Liu, C. Wang, B. Su, and C. Zheng, "Semmae: Semantic-guided masking for learning masked autoencoders," Advances in Neural Information Processing Systems, vol. 35, pp. 14 290–14 302, 2022.

[56] J. Xu, Z. Lin, D. Zhou, Y. Yang, X. Liao, Q. Wang, B. Wu, G. Chen, and P.-A. Heng, "Dppmask: Masked image modeling with determinantal point processes," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024, pp. 2266–2276.

[57] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in Proceedings of the 25th international conference on Machine learning, 2008, pp. 1096–1103.

[58] H. Liu, X. Jiang, X. Li, A. Guo, Y. Hu, D. Jiang, and B. Ren, "The devil is in the frequency: Geminated gestalt autoencoder for self-supervised visual pre-training," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, no. 2, 2023, pp. 1649–1656.

[59] S. Li, D. Wu, F. Wu, Z. Zang, and S. Z. Li, "Architecture-agnostic masked image modeling from vit back to cnn," in International Conference on Machine Learning. PMLR, 2023, pp. 20 149–20 167.

[60] C. Wei, H. Fan, S. Xie, C.-Y. Wu, A. Yuille, and C. Feichtenhofer, "Masked feature prediction for self-supervised visual pre-training," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 14 668–14 678.

[61] C. Huang, Q. Xu, Y. Wang, Y. Wang, and Y. Zhang, "Self-supervised masking for unsupervised anomaly detection and localization," IEEE Transactions on Multimedia, 2022.

[62] Z. Peng, L. Dong, H. Bao, F. Wei, and Q. Ye, "A unified view of masked image modeling," Transactions on Machine Learning Research, 2022.

[63] H. Xue, P. Gao, H. Li, Y. Qiao, H. Sun, H. Li, and J. Luo, "Stare at what you see: Masked image modeling without reconstruction," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 22 732–22 741.

[64] A. El-Nouby, G. Izacard, H. Touvron, I. Laptev, H. Jegou, and E. Grave, "Are large-scale datasets necessary for self-supervised pre-training?" arXiv preprint arXiv:2112.10740, 2021.

[65] X. Liu, J. Zhou, T. Kong, X. Lin, and R. Ji, "Exploring target representations for masked autoencoders," arXiv preprint arXiv:2209.03917, 2022.

[66] K. Tian, Y. Jiang, C. Lin, L. Wang, Z. Yuan et al., "Designing bert for convolutional networks: Sparse and hierarchical masked modeling," in The Eleventh International Conference on Learning Representations, 2022.

[67] P. Gao, T. Ma, H. Li, Z. Lin, J. Dai, and Y. Qiao, "Convmae: Masked convolution meets masked autoencoders," arXiv preprint arXiv:2205.03892, 2022.

[68] J. Liu, X. Huang, J. Zheng, Y. Liu, and H. Li, "Mixmae: Mixed and masked autoencoder for efficient pretraining

of hierarchical vision transformers," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 6252–6261.

[69] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 11 976–11 986.

[70] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," arXiv:1703.06870, Jan. 2018. [Online]. Available: http://arxiv.org/abs/1703.06870

[71] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," arXiv preprint arXiv:1711.05101, 2017.

[72] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch sgd: Training imagenet in 1 hour," arXiv preprint arXiv:1706.02677, 2017.

[73] Y. Fang, S. Yang, S. Wang, Y. Ge, Y. Shan, and X. Wang, "Unleashing vanilla vision transformer with masked image modeling for object detection," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 6244–6253.

[74] Q. Liu, X. Li, Z. He, C. Li, J. Li, Z. Zhou, D. Yuan, J. Li, K. Yang, N. Fan et al., "Lsotb-tir: A large-scale high-diversity thermal infrared object tracking benchmark," in Proceedings of the 28th ACM international conference on multimedia, 2020, pp. 3847–3856.

[75] H. Sun, J. Bai, F. Yang, and X. Bai, "Receptive-field and direction induced attention network for infrared dim small target detection with a large-scale dataset irdst," IEEE Transactions on Geoscience and Remote Sensing, vol. 61, pp. 1–13, 2023.

[76] T. Imaging, "Flir data set dataset," https://universe.roboflow.com/thermal-imaging-0hwfw/flir-data-set, mar 2024, visited on 2024-07-16. [Online]. Available: https://universe.roboflow.com/thermal-imaging-0hwfw/flir-data-set

[77] H. Wang, L. Zhou, and L. Wang, "Miss detection vs. false alarm: Adversarial learning for small object segmentation in infrared images," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 8509–8518.

[78] J. Portmann, S. Lynen, M. Chli, and R. Siegwart, "People detection and tracking from aerial thermal views," in 2014 IEEE international conference on robotics and automation (ICRA). IEEE, 2014, pp. 1794–1800.

[79] J. Suo, T. Wang, X. Zhang, H. Chen, W. Zhou, and W. Shi, "Hit-uav: A high-altitude infrared thermal dataset for unmanned aerial vehicle-based object detection," Scientific Data, vol. 10, no. 1, p. 227, 2023.

[80] M. Zhang, R. Zhang, Y. Yang, H. Bai, J. Zhang, and J. Guo, "Isnet: Shape matters for infrared small target detection," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 877–886.

[81] Z. Huang, X. Jin, C. Lu, Q. Hou, M.-M. Cheng, D. Fu, X. Shen, and J. Feng, "Contrastive masked autoencoders are stronger vision learners," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023.

[82] C. Tao, X. Zhu, W. Su, G. Huang, B. Li, J. Zhou, Y. Qiao, X. Wang, and J. Dai, "Siamese image modeling for self-supervised vision representation learning," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 2132–2141.