LADDER SIAMESE NETWORK: A METHOD AND INSIGHTS FOR MULTI-LEVEL SELF-SUPERVISED LEARNING

Ryota Yoshihashi  Shuhei Nishimura  Dai Yoneyabashi  Yuya Otsuka  Tomohiro Tanaka  Takashi Miyazaki
Yahoo Japan Corporation

ABSTRACT

In Siamese-network-based self-supervised learning (SSL), multi-level supervision (MLS) is a natural extension to enforce intermediate representations’ consistency against data augmentations. Although existing studies have incorporated MLS to boost their system performances in combination with other ideas, vanilla MLS has not been deeply analyzed. Here, we extensively investigate how MLS works and how much impact it has on SSL performance with various training settings to understand the effectiveness of MLS by itself. For this investigation, we develop a simple Siamese-SSL-based MLS framework Ladder Siamese Network, equipped with multi-level, non-contrastive, and global/local self-supervised training losses. We show that the proposed framework can simultaneously improve BYOL baselines in classification, detection, and segmentation solely by adding MLS. In comparison with the state-of-the-art methods, our Ladder-based model achieves competitive and balanced performances in all tested benchmarks without causing large degradation in any of them, which suggests the usability for building a multi-purpose backbone.

Index Terms— Representation learning, self-supervised learning, Siamese network, multi-level supervision

1. INTRODUCTION

Conventional deep neural networks are notoriously label-hungry, requiring massive human-annotated training data to demonstrate their full performance [1]. Self-supervised learning (SSL) [2] is promising to reduce this annotation dependency, and to make machine learning more autonomous toward enabling more human-like learning mechanisms.

Cross-view learning with Siamese networks [2, 3, 4], including contrastive [2, 3] and non-contrastive methods [4, 5], is a promising SSL approach. The Siamese methods share the idea of learning to make pairs of representations from the same instances similar. Multi-level supervision (MLS) can be generally applicable to the Siamese methods; MLS encourages each stage of representations, not only the final one, in the hierarchical networks to be consistent to data augmentations within the same instances, which is seemingly beneficial to enhance intermediate layers’ training progress.

Despite the conceptual simplicity, analyses focusing on the MLS in SSL are rarely conducted, and our understanding of MLS’s behavior is weak. All of the previous work that adopted MLS [6, 7, 8] utilized it in a tie-in by combining MLS with other ideas to boost the system performance. Moreover, the effectiveness of MLS might be underestimated; DetCo [6] sacrificed classification accuracy with the MLS to improve detection performance, rather than improving both.

In contrast to the previous work, we explore the effectiveness of MLS on its own by incorporating it into a simpler Siamese framework. For this purpose, we develop a framework to exploit with Siamese non-contrastive SSL, named Ladder Siamese Network after an autoencoder-based un- and semi-supervised learning method [9] with multi-level losses in the pre-SSL era. Interestingly, by incorporating in BYOL-based [4] non-contrastive Siamese SSL, MLS did not harm classification accuracy and in contrast, improved classification and localization tasks i.e., detection and segmentation simultaneously. As a result, Ladder Siam outperformed all of the existing MLS-based SSL methods [6, 7, 8], and performs competitively to the state-of-art SSL methods while keeping simplicity.

Our analyses further provide the following findings: 1) a dense SSL loss, which is designed to improve dense prediction tasks [10, 11], is less harmful to classification performance when it is used in the lower layers with MLS than in the top level as in the conventional settings. 2) MLS gives stronger invariance and discriminability to the intermediate representation without harming those of the final representation. 3) Lower-level losses provide supervisory signals diversified over both foregrounds and backgrounds, while the top-level loss is attentive to foregrounds even when back-projected onto the lower-level representations.

2. RELATED WORK

Many of the Siamese SSL methods after the pioneer SimCLR [2] have similar overall architecture [5] and explore various loss functions, prediction modules, and network update rules. Typical examples include momentum-based MoCo [3, 12], clustering-based SwAV [13], negative-free BYOL [4]. However, we argue that architectural changes, such as the addition of intermediate losses, have been less investigated.

We are aware of a few studies that adopted multi-level self-supervisions (MLS) in Siamese SSL, DetCo [6] used MLS in combination with local patch-wise contrastive learning, CsMi [7] combined MLS with nearest-neighbor-based positive-pair augmentation. Hierarchical Augmentation Invariance [8] assigned specific data augmentation types for each level to learn invariance against them. Remarkably, all of the work presented MLS in bundles to boost the system performance after combining them with other ideas. We instead focus on the analyses of vanilla MLS and show that even a straightforward implementation based on BYOL can outperform the preceding MLS methods.

A number of Siamese SSL methods incorporate region- or pixel-wise learning, which is useful to improve locality awareness and spatial granularity of representations. Region-based methods often use an extra region-proposal module. For example, DetCon [14] utilizes multiscale combinatorial grouping [15], SoCo [16] and UniVIP [17] utilize selective search [18], and CYBORGs [19] and Odin [20] utilize region grouping by k-means [21] to define region-to-region losses. While they are effective especially in object detection, we explore a simpler method that does not rely on extra modules yet can improve detection and segmentation.
3. METHOD

3.1. Preparation: Siamese SSL
First, we briefly review the Siamese SSL framework [4, 5] as a background and introduce notations. Given an input \( x \), a deep network with \( N \) stages that maps \( x \) to the output \( y \) generally can be written as

\[
\begin{align*}
  y &= f_N(z_{N-1}) \\
  z_i &= f_i(z_{i-1}) & (i = 1, 2, ..., N - 1) \\
  z_0 &= x,
\end{align*}
\]

where \( f_i \) denotes the \( i \)-th stage of the network and \( z_i \) denotes the intermediate representations produced by \( f_i \). Here, stages mean certain groups of layers in networks (i.e., conv1, res2, res3, ... in ResNets [22]), typically grouped by their resolutions and divided by downsampling layers.

Siamese frameworks exploit two views of single instances, which are two versions of input images augmented by random transformation. Given an input \( x \), using its two views \( x^a \), \( x^b \) and their corresponding outputs \( y^a = f(x^a) \), \( y^b = f(x^b) \), a self-supervised loss function is defined as \( L(y^a, y^b) \). The network for the second view \( f \) may be identical to \( f \) [2], or the slow Taylored version of \( f \) with momentum [4].

An example of the concrete form of \( L(y^a, y^b) \) is the mean-squares (MSE) with a predictor, introduced by BYOL [4], which is denoted by

\[
L_{BYOL}(y^a, y^b) = \|q(y^a) - y^b\|^2, \tag{2}
\]

where \( q \) is an multi-layer perceptron (MLP) called a predictor. The output of the predictor is normalized. The predictors are to give the losses asymmetry, which is empirically beneficial for overall performances.

In gradient-based optimization of the loss in Eq. 2, updates of the intermediate layers \( f_{N-1}, f_{N-2}, ..., f_1 \) are purely based on backpropagation, which might be indirect. Our motivation is to expose the intermediate layers directly to their own learning objectives.

3.2. Ladder Siamese Network
To enhance learning of intermediate layers, we add losses on the basis of the intermediate representations. We denote the intermediate representations corresponding to the two views \( x^a \) and \( x^b \) by \( z_{i}^a, z_{i}^b, ..., z_{N-1}^a \) and \( z_{i}^b, z_{i}^a, ..., z_{N-1}^b \) respectively. Using these, the overall loss is defined by

\[
L_{all} = L(y^a, y^b) + \sum_{i=1}^{N-1} w_i L_i(z_i^a, z_i^b), \tag{3}
\]

where \( L \) denotes the final-layer loss and \( L_i \) denotes the \( i \)-th intermediate loss. We introduce weight \( w_i \) to control the balance between the losses. Usually Siamese SSL can be seen as a special case of Ladder Siamese SSL where \( w_1 = w_2 = ... = w_{N-1} = 0 \).

For the concrete form of \( L_i(z_i^a, z_i^b) \), we use an adaptation of Eq. 4 for the intermediate layers, which is defined by

\[
\begin{align*}
  L_i &= \|q_i(y_i^a) - y_i^b\|^2, \tag{4} \\
  y_i^k &= p_i(\text{avgpool}(z_i^k)) \quad (k = a, b).
\end{align*}
\]

3.3. Dense loss for lower-layer supervision
While a naive configuration where all-level losses are set to be the same as Eq. 2 is possible, there is a remaining design space for varying each level loss. We exploit this by putting dense losses for lower (input-side) parts of a network, and global losses for higher (output-side) parts. In literature, dense losses [10, 21, 11] equipped with stronger locality-aware supervisory signals have advantages in object detection and segmentation, while they degrade classification accuracy. Here, our intention is to enhance role division of the lower layers and higher layers. Such differentiation in hierarchical networks may naturally emerge [24, 25], and we aim to enhance it to improve locality awareness without largely sacrificing classification accuracies.

Inspired by DenseCL [10], we newly design the DenseBYOL loss, which is a non-contrastive counterpart of DenseCL designed on the basis of MoCo-style contrastive learning. The purpose of this re-invention is to avoid potential ill effects caused by combining contrastive and non-contrastive losses, and maintain conciseness of our BYOL-based basecase. We define our DenseBYOL loss by

\[
\begin{align*}
  L_{Dense}(y^a, y^b) &= |q^\text{com}(y^a) - \text{align}(y^b; y^b)|^2, \tag{5} \\
  \text{align}(y^b; y^b) &= [y_{b,\text{avg}}(u, v) = \text{argmax}_{u,v}(y_i^b, y_i^b)]_{i,j}, \\
  y^i &= p^\text{com}(z^k) \quad (k = a, b),
\end{align*}
\]

where \( \text{align} \) is a spatial resampling operator that picks up corresponding points \((u, v)\) and their feature vectors \( y_i^b,j \) for every \( y_i^b,j \) on the basis of the cosine similarity function denoted by \( \langle \cdot, \cdot \rangle \). The projector and predictor is replaced by \( p^\text{com}, q^\text{com} \), the \( 1 \times 1 \) convolution-based projector and predictor, which no longer require global pooling. Figure 1b illustrates this dense version of the predictor.

We set the dense losses in the lower half of the network, i.e., res2 and res3 of a ResNet, and the global losses in the others. Following DenseCL [10], we used the dense losses in combination with the global ones by averaging.

4. EXPERIMENTS
We evaluate and analyze Ladder Siam following the standard protocol in prior work [3, 11]; we first pretrained our networks with the proposed method using the ImageNet dataset, and then finetuned them or built classifiers on them as feature extractors with frozen parameters in the downstream tasks.
Table 1. Results of BYOL and our Ladder-BYOL.

| Pretraining epochs | Task | BYOL | Ladder-BYOL |
|--------------------|------|------|-------------|
| 100 epochs         | IN acc@1 | 67.4 | 68.3 (+ 0.9) |
|                    | CC box mAP | 39.3 | 40.5 (+ 1.2) |
|                    | VOC mIoU  | 63.8 | 66.6 (+ 2.8) |
| 200 epochs         | IN acc@1 | 71.7 | 72.8 (+ 1.1) |
|                    | CC box mAP | 40.9 | 41.4 (+ 0.5) |
|                    | VOC mIoU  | 64.3 | 67.4 (+ 3.1) |
| 400 epochs         | IN acc@1 | 73.1 | 73.6 (+ 0.5) |

**Pretraining** We pretrained Ladder Siam on the ImageNet-1k [30] dataset in unsupervised fashion, i.e., without using labels. We followed BYOL [4] in optimizer and scheduling settings. We trained two variations of Ladder Siam. Ladder-BYOL is the simpler one where all intermediate losses are the BYOL-style global loss described in Eq. 2. Ladder-DenseBYOL is the dense-loss-equipped alternative more oriented toward dense-prediction tasks e.g., segmentation; it replaced the intermediate losses on the earlier-half stages by the dense loss described in Eq. 5. For comparisons, we additionally implemented DenseBYOL, which has no intermediate losses but a top-level loss of Eq. 5. As a backbone architecture, we used ResNet50 [22] for compatibility in comparison with other methods. We set the weight losses at res2, res3, res4, and res5 to 1/16, 1/8, 1/4, and 1, respectively, as default.

**Downstream tasks** For classification, we conducted linear probing using ImageNet-1k. Linear classifiers were trained on the frozen representation using SGD. The training of the classifier was done during 100 epochs with cosine annealing. For detection, we finetuned Mask R-CNN [31] with FPN [32] on the COCO dataset [33], train2017 for training and val2017 for evaluation. The training schedule was set to $1 \times$ schedule. For segmentation, we finetuned FCNs [34] on PASCAL VOC [35] and Cityscapes [36]. While we did not see a major consensus among the SSL literature on segmentation-evaluation protocol, we used FCN-D8, which is an FCN modified to have eight-pixel stride by dilated convolutions, provided by mmsegmentation as the simplest option. In PASCAL VOC, we used train$\_augs2012$ for training. We set the input resolution to $512 \times 512$ and training iterations to 20k. In Cityscapes, we used the train$\_fine$ subset for training. We set input resolution to $769 \times 769$ and training iterations to 40k. This setting is the same with that of [3, 11].

**Comparisons with the baseline** Comparisons of our Ladder-BYOL with the baseline BYOL are shown in Table 1. We observed improvements over the baselines with our Ladder version on all datasets and training schedules we used. The relative improvements were 0.9 % points in ImageNet linear classification (IN), 1.2 % points in COCO (CC) detection, and 2.8 % points in VOC segmentation when we adopted 100-epoch pretraining. We also observed the similar improvements with 200-epoch pretraining, which shows that our Ladder Siam training framework is consistently beneficial in combination with BYOL. The improvement is + 0.5 % points in IN with the longer 400-epoch pretraining. This relative improvement is a bit smaller than in shorter-term training, and we regard this as the result of faster convergence.

**Comparisons with state of the art methods** We show the results in Table 2. We selected recent ResNet50-based SSL methods that do not rely on extra region extractors or multi-crop strategies, for which we can draw a fair comparison. We basically reported 200-epoch results of ours and compared models. Incidentally, as marked by $\dagger$ in the table, we placed the classification accuracy of the 400-epoch-pretrained model for PixPro due to the unavailability of 200-epoch weights. We placed classification accuracy of the 100-epoch model for RegionCL-SimSiam, which we expect to be similar to its 200-epoch results due to the fast convergence and saturation of SimSiam-based methods [2].

Given the diversity of the downstream tasks, we do not see a single clear winner. However, our Ladder-BYOL maintain a balance of downstream performances at a high level by being the best in ImageNet-1k (IN) linear classification, the best in COCO (CC) box-based detection, and the second best in instance segmentation. For comparison, LEWEL-BYOL [28] performed well and similarly in classification and detection to ours, but was found to be less generalizable to segmentation. In contrast, DenseCL [10] was the best in VOC segmentation but at the cost of classification accuracy. Ladder-DenseBYOL is the second best in VOC semantic segmentation, while it has similar but slightly worse performances in the other tasks than Ladder-BYOL. Thus, it can be regarded as a still versatile but somewhat segmentation-oriented backbone.

**Component-wise comparisons** We further focus on methods that have connections on the underlying ideas and consist of similar components to ours. First, we compare the effect of adding intermediate losses with DetCo [6]. Table 3 shows the classification-performance changes by adding MLS by the intermediate losses, which were provided by the original paper [6] as a part of an ablative study and computed by us. While the two DetCo variants degraded their classification performances by MLS, ours improved in contrast. A hypothesis on the cause of this reversal is the difference of contrastive and non-contrastive losses; the contrastive loss used in DetCo can be backtracked by its reliance on negative pairs, which are sometimes too hard to distinguish from positives [37]. This might be more harmful when the losses are assigned to less powerful intermediate layers.

Table 4 compares the effect of dense SSL losses [10, 23]. Note that PixPro and DenseCL used dense losses in the top level, but our Ladder models exploited dense losses on intermediate layers. Regardless of base methods or dense loss types, adding the dense losses degraded classification and improved segmentation in all examined conditions. However, classification degradation in our Ladder-DenseBYOL, which is - 0.7 % - points, is softer than in the others, suggesting that dense losses as the intermediate supervision are a reasonable way to relieve classification degradation.

**Ablative analyses** We conducted ablative analyses of the intermediate losses and the results are summarized Table 5. Ladder-BYOL and Ladder-DenseBYOL were confirmed to outperform their single-loss counterparts. We additionally tested a Ladder-DenseBYOL variation where all losses are dense, but it was suboptimal in both classification and segmentation, offering more evidence of the effectiveness of mixing dense and global losses.

**Analyses on intermediate representations** Table 6 summarizes linear probing results of intermediate representations in each stage. Classification improvements in all intermediate layers were confirmed. Figure 2 shows distributions of euclidean distance between two random data-augmented views measured in each level as violin plots. The distance was computed using representation vectors after global average pooling. Ladder training provided stronger consistency against the data augmentation to the intermediate layers lower than res4, which is seemingly the source of the improved intermediate-layer discriminability.

**Gradient visualization** Finally, we investigated whether the roles of each intermediate loss as a supervisor are similar or whether some sort of role divisions emerge. We observed signs of role divisions in Fig. 3, which visualized gradients of each-level loss with reference to an intermediate representation. Given the representation $z_{res2}$ and the losses $L_{res3}$, $L_{res4}$ and $L_{res5}$ viewed as a function on $z_{res2}$, we
Table 2. Performance comparison in various vision tasks by state-of-the-art SSL methods and ours. **Bold** indicates the best results. *†* indicates the second best results. | Method | Classification IN acc. | Detection in COCO box mAP | Detection in COCO mask mAP | Semantic segmentation VOC mIoU | Semantic segmentation Cityscapes mIoU | Avg. rank |
|---|---|---|---|---|---|---|
| ResSim [26] | 66.1 | 40.0 | 36.1 | – | 76.3 | 6.4 |
| DetCo [6] | 68.6 | 40.1 | 36.4 | – | 76.5 | 6.4 |
| DenseCL [10] | 63.3 | 40.3 | 36.4 | 69.4 | 75.7 | 6.4 |
| PixPro [23] | 66.3* | 40.5 | 36.6 | – | 76.3 | 6.4 |
| HAI-SimSiam [27] | 70.1 | – | – | – | – | 6.4 |
| LEWEL-BYOL [28] | 72.8 | 41.3 | 37.4 | 65.7† | 71.3† | 3.4 |
| RegionCL-SimSiam* [29] | 71.3 | 38.8 | 35.2 | 64.8† | 74.1† | 3.4 |
| RegionCL-DenseCL [29] | 68.5 | 40.4 | 36.7 | 64.8† | 74.1† | 3.4 |
| CsMi [7] | 71.6 | 40.3 | 36.6 | – | – | 3.4 |
| DenseSiam [11] | – | 40.8 | 36.8 | – | 77.0 | 3.4 |
| Ladder-BYOL (ours) | 72.8 | 41.4 | 37.2 | 67.4 | 73.9 | 3.4 |
| Ladder-DenseBYOL (ours) | 72.0 | 41.1 | 37.0 | 68.6 | 75.2 | 3.4 |

Table 3. Effect of adding multi-level supervision on classification. |

| Method | res2 | res3 | res4 | res5 | IN acc@1 | VOC mIoU |
|---|---|---|---|---|---|---|
| Baseline | 64.3 | 63.2 (-1.1) | 66.8 (-0.5) | 71.0 | 68.6 | 75.2 |
| DetCo w/o GLS [6] | 67.1 | 66.6 (+0.5) | 71.4 | 72.8 (+1.1) | 68.6 | 75.2 |
| DetCo w/ GLS [6] | 71.7 | 72.8 (+1.1) | 72.8 | 72.8 (+1.1) | 69.0 | 75.4 |

Table 4. Effect of adding dense SSL losses. The results are compared in 100-epoch pretraining. |

| Method | Dense | IN acc@1 | VOC mIoU |
|---|---|---|---|
| Base MoCov2 [12] | 67.6 | 67.5 |
| DenseCL [10] ✓ | 67.4 | 63.8 |
| base BYOL [4] | 66.3 (-1.1) | 65.0 (+1.2) |
| PixPro [23] ✓ | 65.2 (-2.2) | 65.2 (+1.4) |
| DenseBYOL | 72.8 | 67.4 |
| base Ladder-BYOL ✓ | 72.0 (-0.8) | 69.6 (+1.2) |
| Ladder-DenseBYOL | 72.0 (-0.8) | 68.6 (+1.2) |

Table 5. Effect of replacing global losses with dense losses. “G” and “D” denote the usage of global and dense losses respectively. We used 100-epoch pretraining. |

| Method | res2 | res3 | res4 | res5 | IN acc@1 | Voc mIoU |
|---|---|---|---|---|---|---|
| BYOL | – | – | – | G | 67.4 | 64.3 |
| DenseBYOL | – | – | – | D | 65.2 | 65.1 |
| Ladder-B | G | G | G | G | 68.8 | 66.6 |
| Ladder-DB | D | D | G | G | 68.2 | 67.1 |
| Ladder-BYOL (ours) | 28.5 | 40.9 | 56.8 | 68.1 |
| Ladder-DenseBYOL (ours) | 31.9 | 47.2 | 61.7 | 68.7 |

Table 6. Results of classification using intermediate layers. |

| Method | res2 | res3 | res4 | res5 | IN acc@1 | Voc mIoU |
|---|---|---|---|---|---|---|
| BYOL | 28.5 | 40.9 | 56.8 | 68.1 |
| Ladder-BYOL | 31.9 | 47.2 | 61.7 | 68.7 |

computed $\frac{\partial L}{\partial z_{res2}}$, which is the contribution of each $L_{res4}$ to the total gradient of $z_{res2}$ for $i = 3$, 4, 5. We excluded $L_{res2}$ in the visualization because the global average pooling on $z_{res2}$ provides spatially uniform gradients, which is improper for visualization. For plotting, we took the absolute-sum along the channel axis and visualized 2-D patterns of gradient magnitude. In Fig. 3, the later-level gradients are more focused on objects while the earlier-level ones are more globally distributed on both foregrounds and backgrounds, which can be useful to widely collect learnable factors. At the same time, the earlier losses might be non-object-centric when disrupted by background clutters, and here we hypothesize that later-level and earlier-level losses work complementarily together. While the local/global role division of lower and higher layers is well known, our finding is that MLS may enhance it by producing self supervisions that match the roles of each level.

**Runtime** BYOL, Ladder-BYOL, and Ladder-DenseBYOL took 34:58, 35:43, and 35:31 for 100-epoch pretraining respectively. Because their differences are only in the loss computation, the computational overheads were negligible compared with the random variations caused by the environment in real-time measurement.

5. CONCLUSION

In this paper, we presented Ladder Siamese Network, conceptually simple yet effective framework to learn versatile self-supervised representations. Our series of analyses provide insights toward understanding how Siamese SSL can improve lower- to higher-level visual representations. In future, we will explore further effective combinations of loss functions such as region-proposal-based and unsupervised-segmentation-based ones with our Ladder Siamese framework.

Fig. 2. Distribution of euclidean distance between the two data-augmented views measured in each stage of the network.

Fig. 3. Visualization of gradients, i.e., supervisory signals during training at the earliest-stage provided by each intermediate loss.
6. REFERENCES

[1] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta, “Revisiting unreasonable effectiveness of data in deep learning era,” in *ICCV*, 2017, pp. 843–852.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, “A simple framework for contrastive learning of visual representations,” in *ICML*, 2020, pp. 1597–1607.

[3] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick, “Momentum contrast for unsupervised visual representation learning,” in *CVPR*, 2020, pp. 9729–9738.

[4] Jean-Bastien Grill et al., “Bootstrap your own latent—a new approach to self-supervised learning,” *NeurIPS*, vol. 33, pp. 21271–21284, 2020.

[5] Xinlei Chen and Kaiming He, “Exploring simple siamese representation learning,” in *CVPR*, 2021, pp. 15750–15758.

[6] Enze Xie, Jian Ding, Wenhui Wang, et al., “Detco: Unsupervised contrastive learning for object detection,” in *ICCV*, 2021.

[7] Haohang Xu, Xiaopeng Zhang, Hao Li, et al., “Seed the views: Hierarchical semantic alignment for contrastive representation learning,” *IEEE TPAMI*, 2022.

[8] Junbo Zhang and Kaisheng Ma, “Rethinking the augmentation module in contrastive learning: Learning hierarchical augmentation invariance with expanded views,” in *CVPR*, 2022, pp. 16650–16659.

[9] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko, “Semi-supervised learning with ladder networks,” *NeurIPS*, vol. 28, 2015.

[10] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li, “Dense contrastive learning for self-supervised visual pre-training,” in *CVPR*, 2021, pp. 3024–3033.

[11] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy, “Dense siamese network for dense unsupervised learning,” in *ECCV*, 2022.

[12] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He, “Improved baselines with momentum contrastive learning,” *arXiv preprint arXiv:2003.04297*, 2020.

[13] Mathilde Caron, Ishan Misra, Julien Mairal, et al., “Unsupervised learning of visual features by contrasting cluster assignments,” *NeurIPS*, vol. 33, pp. 9912–9924, 2020.

[14] Olivier J Hénaff, Skanda Koppula, Jean-Baptiste Alayrac, et al., “Efficient visual pretraining with contrastive detection,” in *ICCV*, 2021, pp. 10086–10096.

[15] Pablo Arbeláez, Jordi Pont-Tuset, Jonathan T Barron, Ferran Marques, and Jitendra Malik, “Multiscale combinatorial grouping,” in *CVPR*, 2014, pp. 328–335.

[16] Fangyun Wei, Yue Gao, Zhirong Wu, Han Hu, and Stephen Lin, “Aligning pretraining for detection via object-level contrastive learning,” in *NeurIPS*, 2021, vol. 34, pp. 22682–22694.

[17] Zhaowen Li, Yousong Zhu, Fan Yang, et al., “UniVIP: A unified framework for self-supervised visual pre-training,” in *CVPR*, 2022, pp. 14627–14636.

[18] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders, “Selective search for object recognition,” *IJCV*, vol. 104, no. 2, pp. 154–171, 2013.

[19] Renhao Wang, Hang Zhao, and Yang Gao, “CYBORGS: Contrastively bootstrapping object representations by grounding in segmentation,” in *ECCV*. Springer, 2022, pp. 260–277.

[20] Olivier J. Hénaff, Skanda Koppula, Evan Shelhamer, et al., “Object discovery and representation networks,” in *ECCV*. Springer, 2022, pp. 260–277.

[21] Stuart Lloyd, “Least squares quantization in PCM,” *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016, pp. 770–777.

[23] Zhenda Xie, Yutong Lin, Zheng Zhang, et al., “Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning,” in *CVPR*, 2021, pp. 16684–16693.

[24] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy, “Do vision transformers see like convolutional neural networks?,” *NeurIPS*, vol. 34, pp. 12116–12128, 2021.

[25] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang, “Hierarchical convolutional features for visual tracking,” in *ICCV*, 2015, pp. 3074–3082.

[26] Tete Xiao, Colorado J. Reed, Xiaolong Wang, Kurt Keutzer, and Trevor Darrell, “Region similarity representation learning,” in *ICCV*, 2021, pp. 8392–8401.

[27] Kaiming He, Ross Girshick, and Piotr Dollár, “Rethinking ImageNet pre-training,” in *ICCV*, 2019, pp. 4918–4927.

[28] Lang Huang, Shan You, Mingkai Zheng, et al., “Learning where to learn in cross-view self-supervised learning,” in *CVPR*, 2022, pp. 14451–14460.

[29] Yufei Xu, Qiming Zhang, Jing Zhang, and Dacheng Tao, “RegionCL: Can simple region swapping contribute to contrastive learning?,” in *ECCV*, 2022.

[30] Jia Deng, Wei Dong, Richard Socher, et al., “ImageNet: A large-scale hierarchical image database,” in *CVPR*. IEEE, 2009, pp. 248–255.

[31] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, “Mask R-CNN,” in *ICCV*, 2017, pp. 2961–2969.

[32] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, et al., “Feature pyramid networks for object detection,” in *CVPR*, 2017, pp. 2117–2125.

[33] Tsung-Yi Lin, Michael Maire, Serge Belongie, et al., “Microsoft COCO: Common objects in context,” in *ECCV*. Springer, 2014, pp. 740–755.

[34] Jonathan Long, Evan Shelhamer, and Trevor Darrell, “Fully convolutional networks for semantic segmentation,” in *CVPR*, 2015, pp. 3431–3440.

[35] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman, “The PASCAL visual object classes (VOC) challenge,” *IJCV*, vol. 88, no. 2, pp. 303–338, 2010.

[36] Marius Cordts, Mohamed Omran, Sebastian Ramos, et al., “The Cityscapes dataset for semantic urban scene understanding,” in *CVPR*, 2016, pp. 3213–3223.

[37] Alexander C Li, Alexei A Efros, and Deepak Pathak, “Understanding collapse in non-contrastive learning,” in *ECCV*, 2022.