



Article

Heuristic Attention Representation Learning for Self-Supervised Pretraining

Van Nhiem Tran ^{1,2} , Shen-Hsuan Liu ^{1,2}, Yung-Hui Li ^{2,*}  and Jia-Ching Wang ¹

¹ Department of Computer Science and Information Engineering, National Central University, Taoyuan 3200, Taiwan; tvnhiemhmus@cc.ncu.edu.tw (V.N.T.); 109522071@cc.ncu.edu.tw (S.-H.L.); jcw@csie.ncu.edu.tw (J.-C.W.)

² AI Research Center, Hon Hai Research Institute, Taipei 114699, Taiwan

* Correspondence: yunghui.li@foxconn.com; Tel.: +886-2-2268-3466

Abstract: Recently, self-supervised learning methods have been shown to be very powerful and efficient for yielding robust representation learning by maximizing the similarity across different augmented views in embedding vector space. However, the main challenge is generating different views with random cropping; the semantic feature might exist differently across different views leading to inappropriately maximizing similarity objective. We tackle this problem by introducing **Heuristic Attention Representation Learning (HARL)**. This self-supervised framework relies on the joint embedding architecture in which the two neural networks are trained to produce similar embedding for different augmented views of the same image. HARL framework adopts prior visual object-level attention by generating a heuristic mask proposal for each training image and maximizes the abstract object-level embedding on vector space instead of whole image representation from previous works. As a result, HARL extracts the quality semantic representation from each training sample and outperforms **existing** self-supervised baselines on several downstream tasks. In addition, we provide efficient techniques based on conventional computer vision and deep learning methods for generating heuristic mask proposals on natural image datasets. Our HARL achieves +1.3% advancement in the ImageNet semi-supervised learning benchmark and +0.9% improvement in AP₅₀ of the COCO object detection task over the previous state-of-the-art method BYOL. Our code implementation is available for both TensorFlow and PyTorch frameworks.

Keywords: heuristic attention; perceptual grouping; self-supervised learning; visual representation learning; deep learning; computer vision



Citation: Tran, V.N.; Liu, S.-H.; Li, Y.-H.; Wang, J.-C. Heuristic Attention Representation Learning for Self-Supervised Pretraining. *Sensors* **2022**, *22*, 5169. <https://doi.org/10.3390/s22145169>

Academic Editor: Jing Tian

Received: 4 June 2022

Accepted: 7 July 2022

Published: 10 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Visual representation learning has been an extended research area on supervised and unsupervised methods. Most supervised learning models learn visual representations by training with many labeled datasets, then transferring the knowledge to other tasks [1–5]. Most supervised learning frameworks try to tune their parameters such that they maximally compress mapping the particular input variables that preserve the information on the output variables [6–8]. As a result, most deep neural networks fail to generalize and maintain robustness if the test samples are different from the training samples on variant distribution and domains.

The new approaches are self-supervised representation learning to overcome the existing drawbacks of supervised learning [9–15]. These techniques have attracted significant attention for efficient, generalization, and robustness representation learning when transferring learned representation on multiple downstream tasks achieving on-par or even outperforming supervised baselines. Furthermore, self-supervised learning methods overcome the human supervision capability of leveraging the enormous availability of unlabeled data. Despite various self-supervised frameworks, these methods involve certain

forms of the joint embedding architectures of the two branches neural network such as the Siamese network [16]. The neural networks of two branches are usually weights-sharing or different. In the joint embedding self-supervised framework, the common objective is to maximize the agreement between embedding vectors from different views of the same image. However, the biggest challenge is avoiding collapsing to a trivial constant solution, which is that all output embedding vectors are the same. Several strategies to prevent the collapsing phenomenon can be categorized into two main approaches: contrastive learning and non-contrastive learning. Self-supervised contrastive learning [9,17] prevents collapse via negative sample pairs. However, contrastive learning requires a large number of negative samples leading to the requirement of high computational resources. The efficient alternative approach is non-contrastive learning [13,14,18]. These frameworks rely only on positive pairs with a momentum encoder [13] or using an extra neural network on one branch with the block gradient flow [14,18].

Most existing contrastive and non-contrastive objectives are optimized based on the whole image semantic features across different augmented views. However, under this assumption, several challenges exist. First, popular contrastive methods such as SimCLR [9] and MoCo [17] require more computation and training samples than supervised methods. Second, more importantly, there is no guarantee that semantic representation of different objects will differentiate between different cropping views of the same image. For instance, several meaningful objects (vehicles, humans, animals, etc.) may exist in the same image. The semantic representation of vehicles and humans is different, so contrasting the similarity between different views based on the whole-image semantic feature may be misleading. Research in cognitive psychology and neural science [19–22] showed that early visual attention helps humans focus on the main group of important objects. In computer vision, the perceptual grouping principle is used to group visual features into meaningful parts that allow a much more effective learning representation of the input context information [21].

Motivated by perceptual grouping, we proposed the **Heuristic Attention Representation Learning (HARL)** framework that comprises two main components. First, the early attention mechanism uses unsupervised techniques to generate the heuristic mask to extract object-level semantic features. Second, we construct a framework to abstract and maximize similarity object-level agreement (foreground and background) across different views beyond augmentations of the same image [13,18,23]. This approach helps enrich the quantity and quality of semantic representation by leveraging foreground and background features extracted from the training dataset.

We can summarize our main findings and contributions as follows:

1. We introduce a new self-supervised learning framework (HARL) that maximizes the similarity agreement of object-level latent embedding on vector space across different augmented views. The framework implementation is available in the Supplementary Material section.
2. We utilized two heuristic mask proposal techniques from conventional computer vision and unsupervised deep learning methods to generate a binary mask for the natural image dataset.
3. We construct the two novel heuristic binary segmentation mask datasets for the ImageNet ILSVRC-2012 [24] to facilitate the research in the perceptual grouping for self-supervised visual representation learning. The datasets are available to download in the Data Availability Statement section.
4. Finally, we demonstrate that adopting early visual attention provides a diverse set of high-quality semantic features that increase more effective learning representation for self-supervised pretraining. We report promising results when transferring HARL's learned representation on a wide range of downstream vision tasks.

The remainder of this paper is organized as follows. In Section 2, we discussed related works. Section 3 introduces the HARL framework in detail. Section 4.1 briefly describes the implementation of the HARL framework in self-supervised pretraining. Section 4.2

evaluates and benchmarks HARL performance on the ImageNet evaluation, transfers learning to other downstream tasks and compares it to previous state-of-the-art methods. In Section 5, we provide the analysis of the components impacting the performance and understanding of the behavior of our proposed method. Finally, this paper is concluded in Section 6.

2. Related Works

Our method is mostly related to unsupervised visual representation learning methods, aiming to exploit input signals' internal distributions and semantic information without human supervision. The early works focused on several design-solving pretext tasks, and image generation approaches. Pretext tasks focus on the aspects of image restoration such as denoising [25], predicting noise [26], colorization [27,28], inpainting [29], predicting image rotation [30], solving jigsaw puzzles [31] and more [32,33]. However, these methods, the learned representation of neural networks pre-trained on pretext tasks, still failed in generalization and robustness when performed on different downstream tasks. The generative adversarial learning [34–36] and variational auto-encoding [25,37,38] operate directly on pixel space and high-level details for image generations, which require costly computation that may not be essential and efficient for visual representation learning.

Self-supervised contrastive learning. The popular self-supervised contrastive learning frameworks [9,39,40] aim to pull semantic features from different cropping views of the same image while pushing other features away from other images. However, the downside of contrastive methods is that they require a considerable number of negative pairs, leading to significant computation resources and memory footprint. The efficient alternative approach is non-contrastive learning [13,18], which only maximizes the similarity of two views from the same image without contrast to other views from different images.

Self-supervised non-contrastive learning. Distillation learning-based framework [13,18] inspired by knowledge distillation [41] is applied to joint embedding architecture. One branch is defined as a student network, and another is described as a teacher network. The student network is trained to predict the representation of the teacher network; the teacher network's weights are optimized from the student network by a running average of the student network's weights [13] or by sharing with the student's weights and blocking the gradient flow through the teacher network [18]. Non-contrastive frameworks are effective and computationally efficient compared to self-supervised contrastive frameworks [9,17,39].

However, most contrastive or non-contrastive self-supervised techniques maximize similarity agreements of the whole-image context representation of different augmented views. While developing localization attention to separate the semantically features [42,43] by the perceptual grouping of semantic information proved that adopting prior mid-level visible in pretraining gains efficiency for representation learning. The most recent study related to our [39] leveraging visual attention with segmentation obtained impressive results when transferring the learned representation to downstream tasks on object detection and segmentation in multiple datasets. In contrast to our work, previous work employs pixel-level models for contrastive learning, which uses backbones specialized for semantic segmentation and uses different loss functions. It is important to note that the primary work objective is difficult to transfer to other self-supervised frameworks. It also did not investigate the masking feature method or the impact of the dimension and size of the output spatial feature maps on the latent embedding representation, which we will examine next.

3. Methods

In contrastive or non-contrastive learning-based frameworks, HARL object-level objectives are applicable. For example, our study implements a non-contrastive learning framework using an exponential moving average weight parameter of one encoder to another and an extra predictor inspired by BYOL [13]. HARL's objective maximizes the

agreement of the object-level (foreground and background) latent embedding vector across different cropping views beyond augmentations shown in Figure 1.

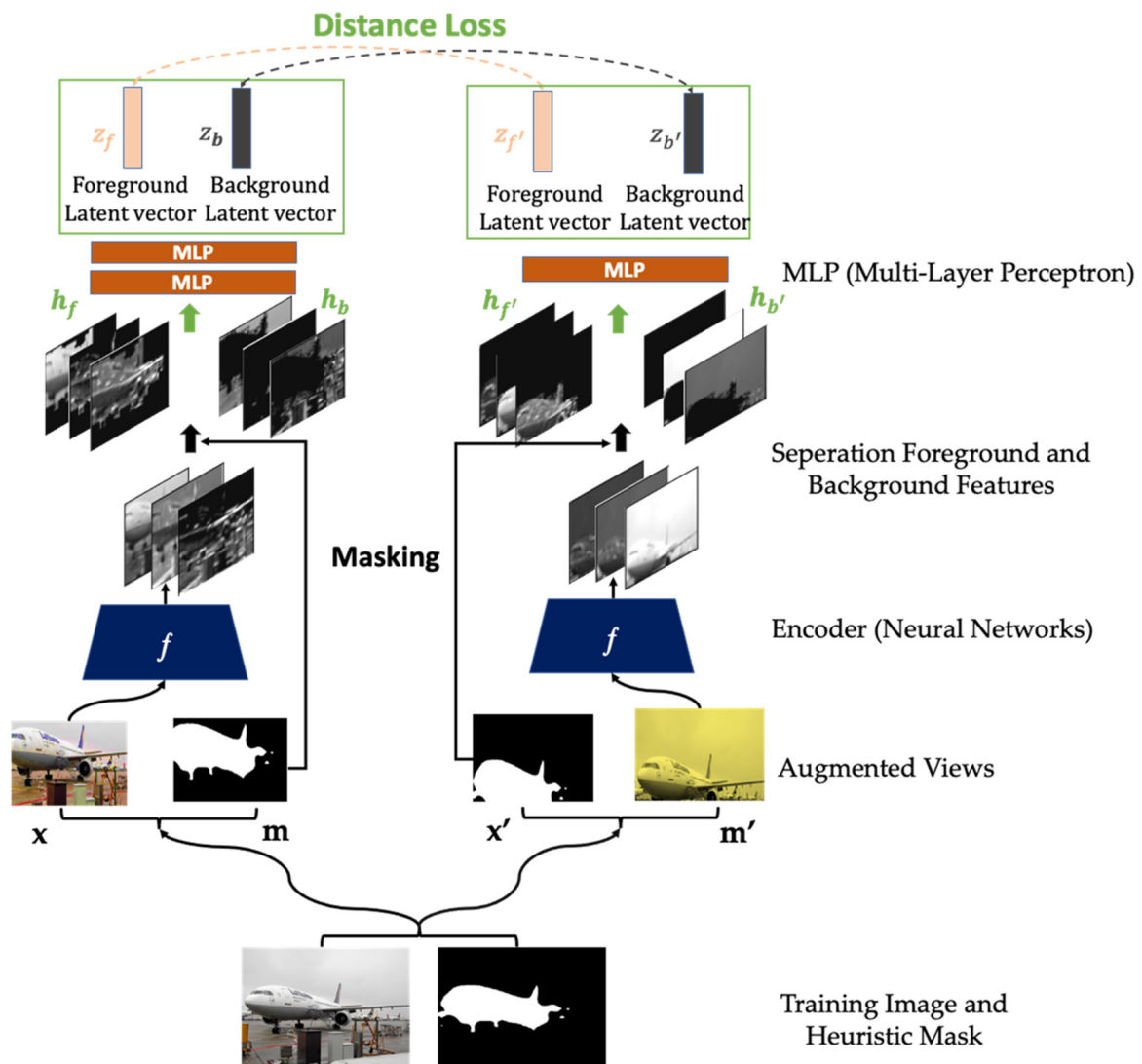


Figure 1. The HARL's architecture. The heuristic binary mask can be estimated by using either conventional computer vision or deep learning approaches. After that, data augmentation transformation is applied to both the image and its mask (**bottom**). Then, the image pairs flow to a convolutional feature extraction module. The heuristic mask is used to mask the feature maps (which are the outputs of the feature extraction module) in order to separate the foreground from the background features (**middle**). These features are further processed by non-linear multi-layer perceptron modules (MLP). Finally, the similarity objective maximizes foreground and background embedding vectors across different augmented views from the same image (**top**).

3.1. HARL Framework

The HARL framework consists of three essential steps. In step 1, we estimate the heuristic binary mask for the input image, which segments an image into foreground and background (see described detail in Section 3.2). Next, these masks can be computed using either conventional computer vision methods such as DRFI [44] or unsupervised deep learning saliency prediction [42]. After the mask is estimated, we perform the same image transformation (cropping, flipping, resizing, etc.) to both the image and its mask. Finally, if it is the RGB image, transformations such as color distortion can be applied to the image, such as the image augmentation pipeline of SimCLR [9]. The detailed augmentation pipeline is described in Appendix A.1. After data augmentation, each image and mask pair

generated two augmented images x, x' aligned with two augmented masks m and m' as illustrated in Figure 1.

In step 2, we implement standard ResNet-50 [45] convolution residual neural network for feature extractor denotation as f . Each image through the feature extractor encodes the output to obtain the spatial feature maps of size $7 \times 7 \times 2048$, and this feature extraction process can be formulated as $h = f(x)$, where $h \in \mathbb{R}^{H \times W \times D}$. Then, the feature maps can be separated into the foreground and background feature maps by performing element-wise multiplication with the heuristic binary mask. In addition, we provide ablation studies to analyze the impact of the spatial feature map in various sizes and dimensions, as described in Section 5.1. The foreground and background features are denoted as, h_f, h_b (Appendix A.2 provides detail of the masking feature method). The foreground and background spatial features are down-sampled using global average pooling to project to a smaller dimension with non-linear multi-layer perceptron (MLP) architecture g .

HARL framework structure adapts from BYOL [13], in which one augmented image (x) is processed with the encoder f_θ , and projection network g_θ , where θ is the learned parameters. Another augmented image (x') is processed with f_ξ and g_ξ , where ξ is an exponential moving average of θ . The first augmented image is further processed with the predictor network q_θ . The projection and predictor network architectures are the same using the non-linear multi-layer perceptron (MLP), as detailed in Section 4. The definition of encoder, projection, and prediction network is adapted from the BYOL. Finally, the latent representation embedding vectors corresponding to the augmented image's foreground and background features are denoted as $z_f, z_b, z_{f'},$ and $z_{b'} \in \mathbb{R}^d$.

$$\text{where :} \quad \begin{aligned} z_f, z_b &\triangleq g_\theta \circ q_\theta(h_f, h_b), \\ z_{f'}, z_{b'} &\triangleq g_\xi(h_{f'}, h_{b'}). \end{aligned}$$

In step 3, we compute the HARL's loss function of the given foreground and background latent representations ($z_f, z_b, z_{f'}$ and $z_{b'}$ are extracted from two augmented images x, x') which is defined as mask loss, as illustrated in Equation (1). We apply ℓ_2 -normalization to these latent vectors, then minimize their negative cosine similarity agreement with the weighting coefficient α . We study the impact of α value and the combination of the whole image and object-level latent embedding vector in the loss objective provided in Section 5.2.

$$\mathcal{L}_\theta^{\text{Maskloss}} = - \left(\alpha \cdot \frac{z_f}{\|z_f\|_2} \cdot \frac{z_{f'}}{\|z_{f'}\|_2} + (1 - \alpha) \cdot \frac{z_b}{\|z_b\|_2} \cdot \frac{z_{b'}}{\|z_{b'}\|_2} \right), \quad (1)$$

where $\|\cdot\|_2$ is ℓ_2 -norm, and it is equivalent to the mean squared error of ℓ_2 -normalized vectors. The weighting coefficient α is in the range [0–1].

We symmetrized loss \mathcal{L} by separately feeding augmented image and mask of view one to the online network and augmented image and mask of view two to the target network and vice versa to compute the loss at each training step. We perform a stochastic optimization step to minimize the symmetrized loss $\mathcal{L}_{\text{symmetrized}} = \mathcal{L} + \mathcal{L}^\sim$.

$$\mathcal{L}_{\text{symmetrized}} = \mathcal{L}_\theta^{\text{Maskloss}} + \mathcal{L}_\theta^{\sim\text{Maskloss}}. \quad (2)$$

After pretraining processing is complete, we only keep the encoder θ and discard all other parts of the networks. The whole training procedure summary is in the python pseudo-code Algorithm 1.

Algorithm 1: HARK: Heuristic Attention Representation Learning

Input:

D, M, T , and T' : set of images, mask and distributions of transformations
 $\theta, f_\theta, g_\theta$, and Q_θ : initial online parameters, encoder, projector, and predictor
 ζ, f_ζ, g_ζ ; // initial target parameters, target encoder, and target projector
 Optimizer; // optimizer, updates online parameters using the loss gradient
 K and N ; // total number of optimization steps and batch size
 $\{T_k\}_{k=1}^K$ and $\{\eta_k\}_{k=1}^K$; // target network update schedule and learning rate
 schedule

1. For $k = 1$ to K do
2. $B \leftarrow \{x_i \sim D\}_{i=1}^N$; // sample a batch of N images
3. $C \leftarrow \{m_i \sim M\}_{i=1}^N$; // sample a batch of N mask
4. For $x_i \in B, m_i \in C$
5. $h \leftarrow f_\theta(t(x_i))$; // compute the encoder feature map
6. $h' \leftarrow f_\theta(t'(x_i))$; // compute the target encoder feature map
7. $h_f, h_b \leftarrow m_i * h$; // separate the feature map
8. $h_{f'}, h_{b'} \leftarrow m_i * h'$; // separate the target feature map
9. $z_f, z_b \leftarrow q_\theta(g_\theta(h_f, h_b))$; // compute projections
10. $z_{f'}, z_{b'} \leftarrow g_\zeta(h_{f'}, h_{b'})$; compute target projections
11. $l_i \leftarrow -2 \cdot \left(\alpha \cdot \frac{z_f}{\|z_f\|_2} \cdot \frac{z_{f'}}{\|z_{f'}\|_2} + (1 - \alpha) \cdot \frac{z_b}{\|z_b\|_2} \cdot \frac{z_{b'}}{\|z_{b'}\|_2} \right)$; // compute loss
12. End for
13. $\delta\theta \leftarrow \frac{1}{N} \sum_{i=1}^N \partial l_i$ // compute the total loss gradient w.r.t. θ
14. $\theta \leftarrow \text{optimizer}(\theta, \delta\theta, \eta_k)$; // update online parameters
15. $\zeta \leftarrow \tau_k \zeta + (1 - \tau_k)\theta$; // update target parameters encoder f_θ

3.2. Heuristic Binary Mask

Our heuristic binary mask estimation technique does not rely on external supervision, nor is it trained with the limited annotated dataset. We proposed two approaches using conventional computer vision and unsupervised deep learning to carry it out, and these methods appear to be well generalized for various image datasets. First, we use the traditional computer vision method DRFI [44] to generate a diverse set of binary masks by varying the two hyperparameters (the Gaussian filter variance σ and the minimum cluster size s). In our implementation, we defined $\sigma = 0.8$ and $s = 1000$ for generating binary masks in the ImageNet [24] dataset. In the second approach, we leverage the self-supervised encoder feature extractor of the pre-trained ResNet-50 backbone from [9,42], then pass the output feature maps into a 1×1 convolutional classification layer for saliency prediction. The classification layer predicts the saliency or “foregroundness” of a pixel. Therefore, we take the output values of the classification layer and set a threshold of 0.5 to decide which pixels belong to the foreground. Pixel saliency values greater than the threshold are determined as foreground objects. Figure 2 shows the example heuristic mask estimated by these two methods. The detailed implementation of the two methods, DRFI and deep learning feature extractor combined with 1×1 convolutional layer is described in Appendix C. In most of our experiments, we used the mask generated by the deep learning method because it is faster than DRFI by running with GPU instead of only with CPU.

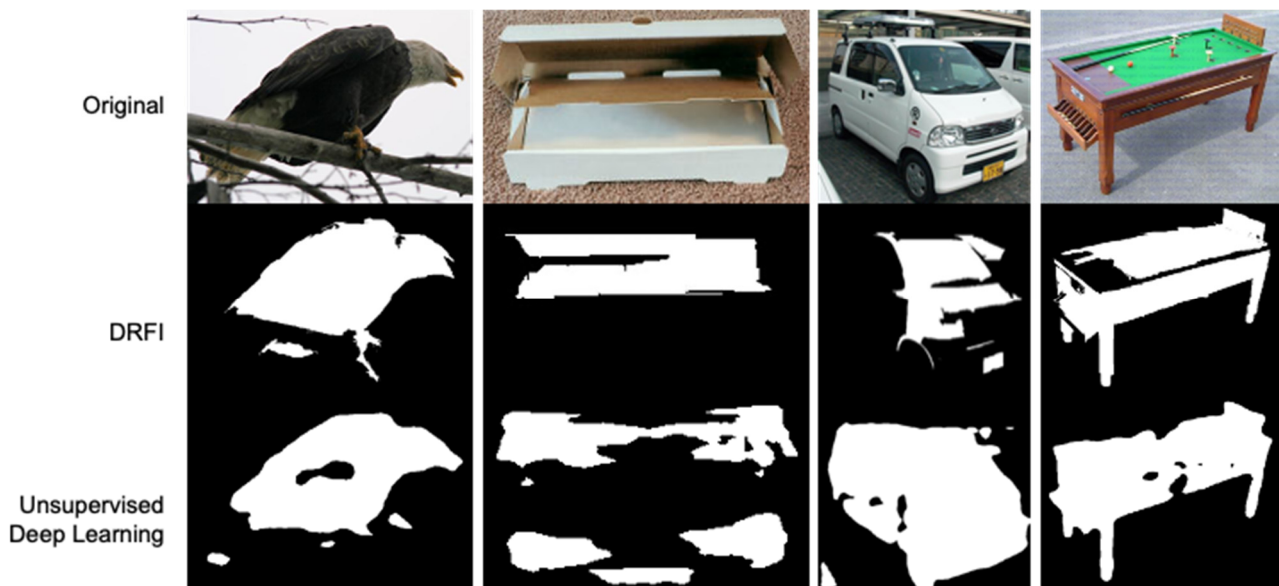


Figure 2. Example of heuristic binary masks used for mask contrastive learning framework. First row: random images from the ImageNet [24] training set. Second row: mask generated based on DRFI algorithm with a predefined sigma σ value of 0.8 and component size values of 1000. The third row is the mask obtained from the self-supervised pre-trained feature extractor ResNet-50 backbone directly followed by a 1×1 convolutional classification foreground and background prediction.

4. Experiments

4.1. Self-Supervised Pretraining Implementation

HARL is trained on RGB images and the corresponding heuristic mask of the ImageNet ILSVRC-2012 [24] training set without labels. We implement standard encoder ResNet [45]. According to previous works by SimCLR and BYOL [9,13], the encoder representation output is projected into a smaller dimension using a multi-layer perceptron (MLP). In our implementation, the MLP comprised a linear layer with an output size of 4960 followed by batch normalization [46], rectified linear units (ReLU) [47] and the final linear layer with 512 output units. We apply the LARS optimizer [48] with the cosine decay learning rate schedule without restarts [49], over 1000 epochs on the base learning rate of 0.2, scaled linearly [50] with the batch size ($\text{LearningRate} = 0.2 \times \text{BatchSize}/256$) and the warmup epochs of 10. Furthermore, we apply a global weight decay parameter of 5×10^{-7} while excluding the biases and normalization parameters from the LARS adaptation and weight decay. The optimization of the online network and target network follow the protocol of BYOL [13]. We use a batch size of 4096 splits over 8 Nvidia A100GPUs. This setup takes approximately 149 h to train a ResNet-50 ($\times 1$).

The computational self-supervised pretraining stage requirements are largely due to forward and backward passes through the convolutional backbone. For the typical ResNet-50 architecture applied to 224×224 resolution images, a single forward pass requires approximately 4B FLOPS. The projection head MLP ($2048 \times 4096 + 4096 \times 512$) requires roughly 10M FLOPS. In our implementation, the convolution network backbone and MLP network are similar compared to baselines BYOL. Since we forward to the foreground and background representation through the projection head two times instead of one, it results in an additional 10M FLOPS in our framework, less than 0.25% of the total. Finally, the cost of computing the heuristic mask images is negligible because they can be computed once and reused throughout training. Therefore, the complexity of each iteration between our method and the baseline BYOL is almost the same for “computational cost” and “training time”.

4.2. Evaluation Protocol

We evaluate the learned representation from the self-supervised pretraining stage on various natural image datasets and tasks, including image classification, segmentation and object detection. First, we assess the obtained representation on the linear classification and semi-supervised learning on the ImageNet following the protocols of [9,51]. Second, we evaluate the generalization and robustness of the learned representation by conducting transfer learning to other natural image datasets and other vision tasks across image classification, object detection and segmentation. Finally, in Appendix B, we provide a detailed configuration and hyperparameters setting of the linear and fine-tuning protocol in our transfer learning implementation.

4.2.1. Linear Evaluation and Semi-Supervised Learning on the ImageNet Dataset

The evaluation for linear and semi-supervised learning follows the procedure in [9,52,53]. For the linear evaluation, we train a linear classifier on top of the frozen encoder representation and report Top-1 and Top-5 accuracies in percentage for the test set, as shown in Table 1. We then evaluate semi-supervised learning, which is fine-tuning the pre-trained encoder on a small subset with 1% and 10% of the labeled ILSVRC-2012 ImageNet [24] training set. We also report the Top-1 and Top-5 accuracies for the test set in Table 1. HARL obtains 54.5% and 69.5% in Top-1 accuracy for semi-supervised learning using the standard ResNet-50 ($\times 1$). It represents a +1.3% and +0.7% advancement over the baseline framework BYOL [13] and significant improvement compared to the strong supervised baseline in the accuracy metric.

Table 1. Evaluation on the ImageNet. The linear evaluation and semi-supervised learning with a fraction (1% and 10%) on ImageNet labels report Top-1 and Top-5 accuracies (in%) using the pre-trained ResNet-50 backbone. The best result is bolded.

Method	Linear Evaluation		Semi-Supervised Learning			
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
Supervised	76.5	-	1%	10%	1%	10%
PIRL [11]	63.6	-	25.4	56.4	48.4	80.4
SimCLR [9]	69.3	89.0	-	-	57.2	83.8
MoCo [17]	60.6	-	48.3	65.6	75.5	87.8
MoCo v2 [54]	60.6	-	-	-	-	-
SimSiam [18]	71.1	-	-	-	-	-
BYOL [13]	71.3	-	-	-	-	-
HARL (ours)	74.3	91.6	53.2	68.8	78.4	89.0
	74.0	91.3	54.5	69.5	79.2	89.3

4.2.2. Transfer Learning to Other Downstream Tasks

We evaluated the HARL's quality of representation learning on linear classification and fine-tuned model following the evaluation setup protocol [9,13,39,55] as detailed in Appendix B.2. HARL's learned representation can perform well for all six different natural distribution image datasets. It has competitive performance in various distribution datasets compared to baseline BYOL [13] and improves significantly compared to the SimCLR [9] approach over six datasets, as shown in Table 2.

We further evaluated HARL's generalization ability and robustness with different computer vision tasks, including object detection of VOC07 + 12 [56] using Faster R-CNN [57] architecture with R50-C4 backbone and instance segmentation task of COCO [58] using Mask R-CNN [59] with R50-FPN backbone. The fine-tuning setup procedure and setting hyperparameter are detailed in Appendix B.3. We report the performance of the standard AP, AP₅₀ and AP₇₅ metrics in Table 3. HARL outperforms the baselines BYOL [13] and also has a significantly better performance than other self-supervised frameworks such as SimCLR [9], MoCo_v2 [17] and supervised baseline on object detection and segmentation.

Table 2. Transfer via fine-tuning on the image classification task. The transfer learning performance between HARL framework and other self-supervised baseline benchmarks across six natural image classification datasets with the self-supervised pre-trained representation on the ImageNet 1000 classes using the standard ResNet-50 backbone. The best result is bolded.

Method	Food101	CIFAR10	CIFAR100	SUN397	Cars	DTD
Linear evaluation:						
HARL (ours)	75.0	92.6	77.6	61.4	67.3	77.3
BYOL [13]	75.3	91.3	78.4	62.2	67.8	75.5
MoCo v2 (repo)	69.2	91.4	73.7	58.6	47.3	71.1
SimCLR [9]	68.4	90.6	71.6	58.8	50.3	74.5
Fine-tuned:						
HARL (ours)	88.0	97.6	85.6	64.1	91.1	78.0
BYOL [13]	88.5	97.4	85.3	63.7	91.6	76.2
MoCo v2 (repo)	86.1	97.0	83.7	59.1	90.0	74.1
SimCLR [9]	88.2	97.7	85.9	63.5	91.3	73.2

Table 3. Transfer learning to other downstream vision tasks. Benchmark the transfer learning performance between HARL framework and other self-supervised baselines on object detection and instance segmentation task. We use Faster R-CNN with C4 backbone for object detection and Mask-RCNN with FPN backbone for instance segmentation. Object detection and instance segmentation backbone initialize with the pre-trained ResNet-50 backbone on ImageNet 1000 classes. The best result is bolded.

Method	Object Detection						Instance Segmentation		
	VOC07 + 12 Detection			COCO Detection			COCO Segmentation		
	AP ₅₀	AP	AP ₇₅	AP ₅₀	AP	AP ₇₅	AP ₅₀ ^{mask}	AP ^{mask}	AP ₇₅ ^{mask}
Supervised	81.3	53.5	58.8	58.2	38.2	41.2	54.7	33.3	35.2
SimCLR-IN [18]	81.8	55.5	61.4	57.7	37.9	40.9	54.6	33.3	35.3
MoCo [17]	82.2	57.2	63.7	58.9	38.5	42.0	55.9	35.1	37.7
MoCo v2 [54]	82.5	57.4	64.0	-	39.8	-	-	36.1	-
SimSiam [18]	82.4	57.0	63.7	59.3	39.2	42.1	56.0	34.4	36.7
BYOL [13]	-	-	-	-	40.4	-	-	37.0	-
BYOL (repo)	82.6	55.5	61.9	61.2	40.2	43.9	58.2	36.7	39.5
HARL (ours)	82.7	56.3	62.4	62.1	40.9	44.5	59.0	37.3	40.0

5. Ablation and Analysis

We study the HARL's components to give the intuition of its behavior and impact on performance. We reproduce the HARL framework with multiple running experiments. For this reason, we hold the same set of hyperparameter configurations and change the configuration of the corresponding component, which we try to investigate. We perform our ablation experiments on the ResNet-50 and ResNet-18 architecture on the ImageNet training set without labels. We evaluate the learned representation on the ImageNet linear evaluation during the self-supervised pretraining stage. To do so, we attach the linear classifier on top of the base encoder with the block gradient flow on the linear classifier's input, which stops influencing and updating the encoder with the label information (a similar approach to SimCLR [9]). We run ablations over 100 epochs and evaluate the performance of the public validation set of the original ILSVRC2012 ImageNet [24] in the Top-1 accuracy metric at every 100 or 200 steps per epoch following the protocol as described in Appendix B.1.

5.1. The Output of Spatial Feature Map (Size and Dimension)

In our HARL framework, separating foreground and background features from the output spatial feature map is essential to maximize the similarity objective across different augmented views. To verify this hypothesis, we analyze several spatial outputs in various

sizes and dimensions by modifying the ResNet kernel's stride to generate the different feature map sizes with the same dimension. For illustration, the standard ResNet is the sequence of four convolution building blocks (conv2_x, conv3_x, conv4_x, conv5_x). For ResNet-50 architecture, the dimension of conv_5x block output feature map is $7 \times 7 \times 2048$. After changing the kernel stride of the conv_4x block from two to one, its new dimension will be $14 \times 14 \times 2048$. In this modified ResNet-50 architecture, the conv5_x block's spatial feature map size is the same as the conv4_x block output.

We conduct the experiment for three different sizes including a deep ResNet-50 ($7 \times 7 \times 2048$, $14 \times 14 \times 2048$, $28 \times 28 \times 2048$) and a shallow ResNet-18 ($7 \times 7 \times 512$, $14 \times 14 \times 512$, $28 \times 28 \times 512$). Figure 3 shows the experimental results of various output sizes and dimensions in the pretraining stage that impact the learned representation when evaluating transfer representation on the ImageNet with linear evaluation protocol. Both shallow and deep ResNet architecture yields better learning ability on the larger output spatial feature map size 14×14 than 7×7 . In our experiments, the performance decreases as we continue to go to a larger output size, 28×28 or 56×56 .

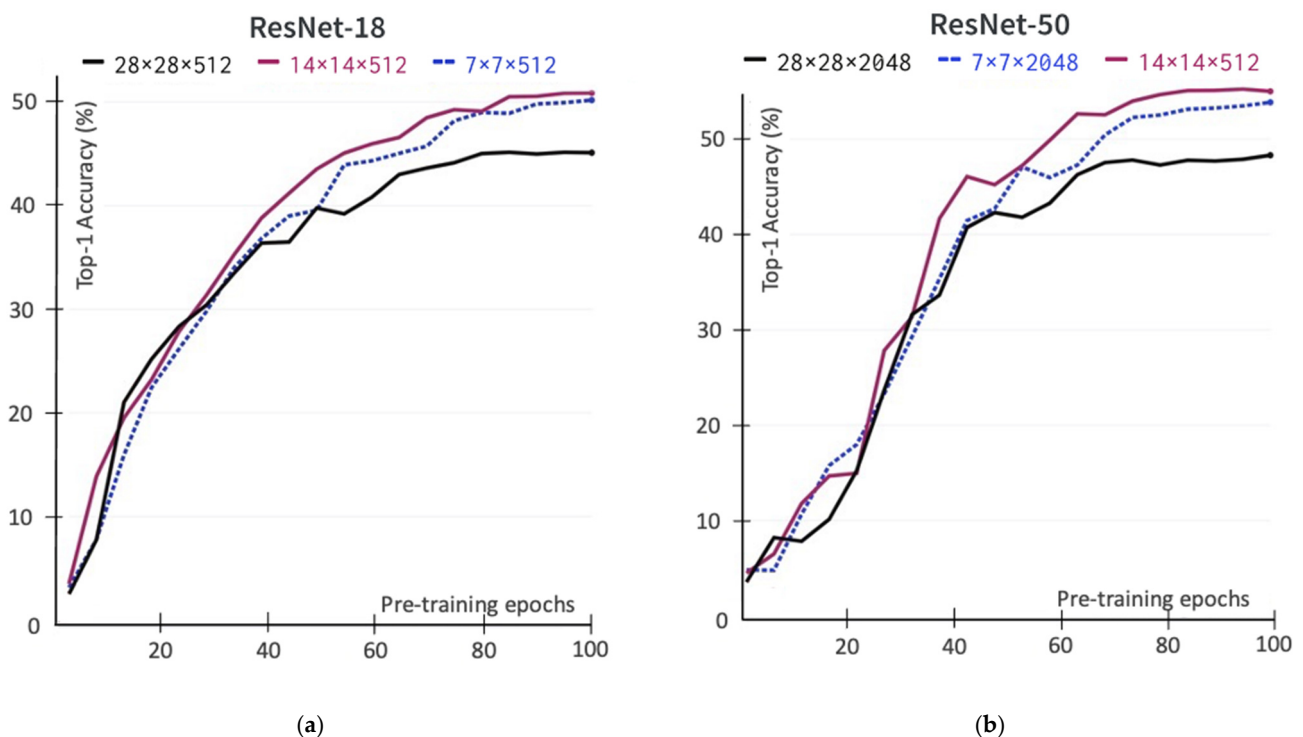


Figure 3. The ImageNet linear evaluation Top-1 accuracy (in %) of spatial output feature maps in various sizes and dimensions during the self-supervised pretraining stage. (a) The self-supervised pre-trained encoder uses the ResNet-18 backbone; (b) The self-supervised pre-trained encoder uses the ResNet-50 backbone.

5.2. Objective Loss Functions

HARL framework structure reuses elements of BYOL [13]. We use two neural networks denoted as *online network* and *target network*. Each network is defined by a set of parameters θ and ζ . The optimization objective minimizes the loss $\mathcal{L}_{\theta, \zeta}$ with respect to learnable parameters θ , while the set ζ is parameterized by using an exponential moving average of the θ , as shown in Equation (3):

$$\zeta \leftarrow \tau \zeta + (1 - \tau) \theta. \quad (3)$$

Unlike previous approaches that minimize loss function only based on the whole image latent embedding vector between two augmented views, HARL minimized the similarity of object-level latent representation, which associated the same spatial regions

abstracting from segmentation mask and thus same semantic meanings. As shown in Figure 1, we use the mask information to separate the spatial semantic object-level feature (foreground and background) of the two augmented views. Then, we minimize their negative cosine similarity, denoting mask loss in Equation (1). In addition to our mask loss objective, we combine the distance loss of the whole image representation and object-level, resulting in hybrid loss as described in Equation (4). We study these two loss objectives in the self-supervised pretraining stage and then evaluate the obtained representation on the ImageNet with a linear evaluation protocol.

5.2.1. Mask Loss

The mask loss objective converges to minimizing the distance loss objective between foreground and background latent embedding on vector space $\mathcal{L}_{\text{foreground}}(\theta, \xi)$ and $\mathcal{L}_{\text{background}}(\theta, \xi)$ with the weighting coefficient α as described in Equation (1). We study the impact of α when it is set to a few predefined values and when it varies according to the cosine scheduling rule. In the first approach, we perform self-supervised pretraining sweeping over three different values $\{0.3, 0.5, 0.7\}$. In the second approach, we schedule the α based on a cosine schedule, $\alpha \triangleq (1 - (1 - \alpha_{\text{base}})) \cdot (\cos \pi k / K) + 1) / 2$, to gradually increase from the starting α_{base} value to 1 corresponding current training step k over total training step K . We tried three α_{base} values, including 0.3, 0.5 and 0.7. We report the Top-1 accuracy on the ImageNet linear evaluation set during the self-supervised pretraining stage, as shown in Figure 4. The weighting coefficient α value of 0.7 yields the consistent learned representation of both approaches. Furthermore, the experimental results demonstrate that the foreground is more important than the background latent representation. For example, in the ImageNet training set, many images exist in which the background information is more than 50% of the image.

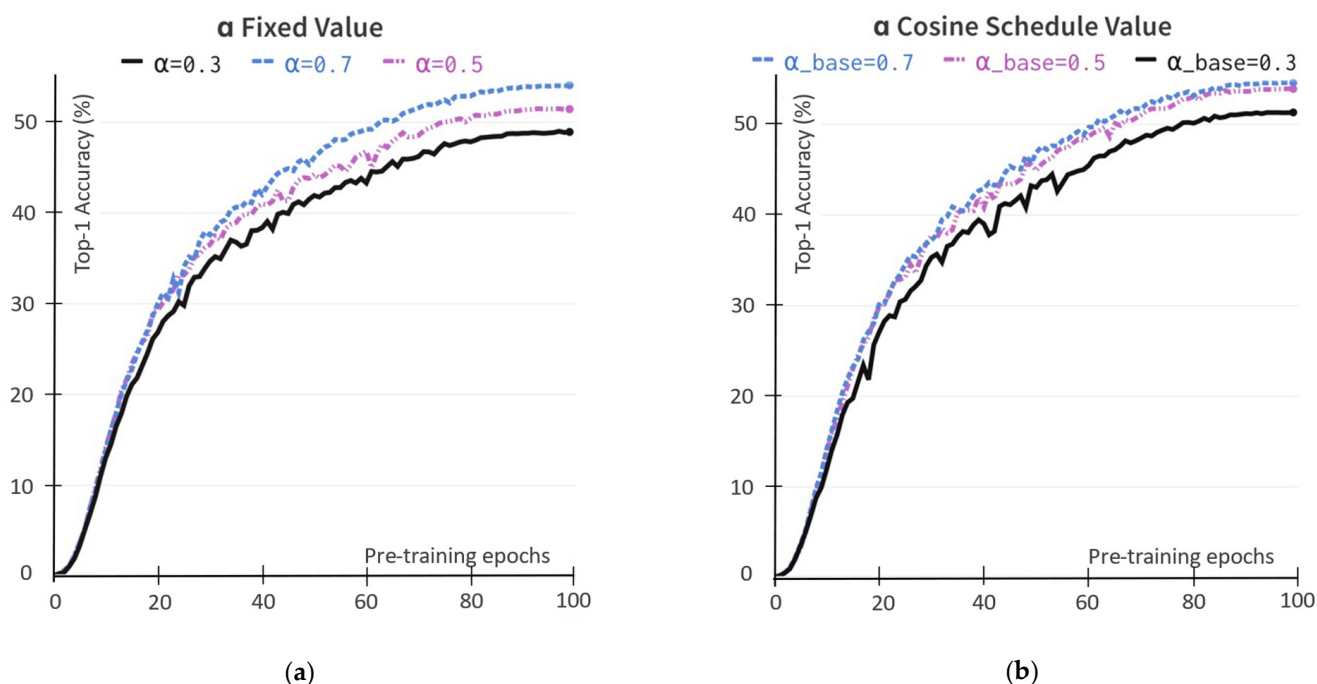


Figure 4. The impact of weighting coefficient α value to the obtained representation during the self-supervised pretraining stage with the ResNet-50 backbone. The evaluation during pretraining uses the ImageNet linear evaluation protocol in Top-1 accuracy (in%). (a) The α value is the fixed value; (b) The α value follows the cosine function scheduler.

5.2.2. Hybrid Loss

The objective combines whole image representation embedding v_1 and v_2 together with object-level representation embedding mask loss described in Equation (1). v_1 and v_2

are extracted from the two augmented views x and x' and are denoted as $v_1 \triangleq_{\theta} g_{\theta} \circ q_{\theta}(x) \in \mathbb{R}^d$ and $v_2 \triangleq_{\theta} g_{\theta} \circ q_{\theta}(x') \in \mathbb{R}^d$. The hybrid loss minimizes the negative cosine similarity with weighting coefficient λ :

$$\mathcal{L}_{\theta}^{hybrid} = - \left[\lambda \cdot \frac{v_1}{\|v_1\|_2} \cdot \frac{v_2}{\|v_2\|_2} + (1 - \lambda) \cdot \mathcal{L}_{\theta}^{Maskloss} \right], \quad (4)$$

where v_1 and v_2 are the whole image latent representation; $\mathcal{L}_{\theta}^{Maskloss}$ is the distance loss computed from the foreground and background latent representation described in Equation (1); $\|\cdot\|_2$ is ℓ_2 -norm; and λ is the weighting coefficient in the range [0–1].

To study the impact of weighting coefficient λ , we use a cosine scheduling value similar to α in the mask loss section. In our experiment, the weighting coefficient λ cosine scheduling sweeping over four λ_{base} values {0.3, 0.6, 0.7, 0.9}. We report the Top-1 accuracy of the ImageNet linear evaluation protocol on the validation set during the self-supervised pretraining stage, shown in Figure 5. We found using the weighting coefficient λ_{base} value of 0.7 obtains the consistent learned representation when transferring to downstream tasks.

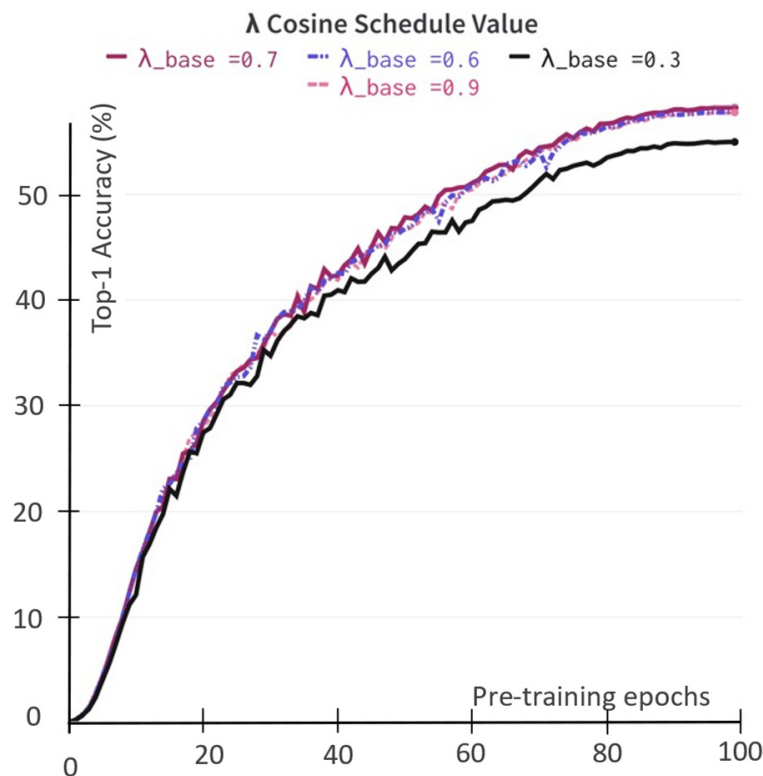


Figure 5. The impact of the weighting coefficient λ value to the obtained representation of the pre-trained encoder (ResNet-50) during the self-supervised pretraining stage on the ImageNet linear evaluation protocol in Top-1 accuracy (in%).

5.2.3. Mask Loss versus Hybrid Loss

We compare the obtained representation using mask loss and hybrid loss on self-supervised pretraining. To do so, we implement the HARL framework with both loss objectives on self-supervised pretraining. We use the cosine schedule function to control the weighting coefficient α and λ sweeping on three different initial values {0.3, 0.5, 0.7} for both coefficients. We evaluate the obtained representation of the pre-trained encoder using ResNet-50 backbone in Top-1 and Top-5 accuracy (in%) on ImageNet linear evaluation protocol, as shown in Table 4. According to the experimental result, using the hybrid loss incorporated between global and object-level latent representation yields better representation learning during self-supervised pretraining.

Table 4. The comparison obtained representation of HARL framework using mask loss and hybrid loss objective. We report Top-1 and Top-5 (in %) accuracy on ImageNet linear evaluation from 100 epochs pre-trained ResNet-50 backbone on ImageNet 1000 classes.

Method	Top-1 Accuracy	Top-5 Accuracy
Mask Loss		
$\alpha_{\text{base}} = 0.3$	51.3	77.4
$\alpha_{\text{base}} = 0.5$	53.9	79.4
$\alpha_{\text{base}} = 0.7$	54.6	79.8
Hybrid Loss		
$\lambda_{\text{base}} = 0.3$	55.0	79.4
$\lambda_{\text{base}} = 0.5$	57.8	81.7
$\lambda_{\text{base}} = 0.7$	58.2	81.8

5.3. The Impact of Heuristic Mask Quality

In our work, the HARL objective uses two different image segmentation techniques. Which ones lead to the best representation? We first consider the heuristics mask retrieving from the computer vision DRFI [44] approach by varying the two hyperparameters (the Gaussian filter variance σ and the minimum cluster size s) as described in detail in Appendix C.1. In our implementation, we generate a diverse set of binary masks by different combinations of $\sigma \in \{0.2, 0.4, 0.8\}$ and $c \in \{1000, 1500\}$. The sets of the generated masks are shown in Figure 6. We found that the setting of $\sigma = 0.8$ and $s = 1000$ generate more stable mask quality than other combinations. Following the deep learning technique, we use the pre-trained deep convolution neural network as the feature extractor and design a saliency head prediction on top of the feature extractor output's representation in the following three steps described in Appendix C.2. The generated masks are dependent on the pixel saliency threshold, which determines the foregroundness and backgroundness of the pixel. In our implementation, we tested the saliency threshold value ranging in $\{0.4, 0.6, 0.7\}$ as shown in Figure 7. We choose the threshold value equal to 0.5 for generating masks in the ImageNet dataset. After choosing the best configure of the two techniques, we generate the mask for the whole training set of the ImageNet [24] dataset. We evaluate the mask quality generated by computing the mean Intersection-Over-Union (mIoU) between masks generated with the ImageNet ground-truth mask annotated by humans from Pixel-ImageNet [60]. The mIoU of the deep learning masks achieves 0.485 over 0.398 of DRFI masks on the subset of 0.485 million images (946/1000 classes of ImageNet). We found that in a complex scene, where multiple objects exist in a single image, the mask generated from the DRFI technique is noisier and less accurate than the deep learning masks, as illustrated in Figure 8.

To fully evaluate the impact of representation learning on downstream performance, we inspect the obtained representational quality with the transfer learning performance on the object detection and segmentation shown in Table 5. The result indicates that for most object detection and segmentation tasks, HARL learning based on masks with deep learning outperforms the one with DRFI masks, although the difference is very small. It shows that the quality of the mask used for HARL does have a small impact on the performance of the downstream task.

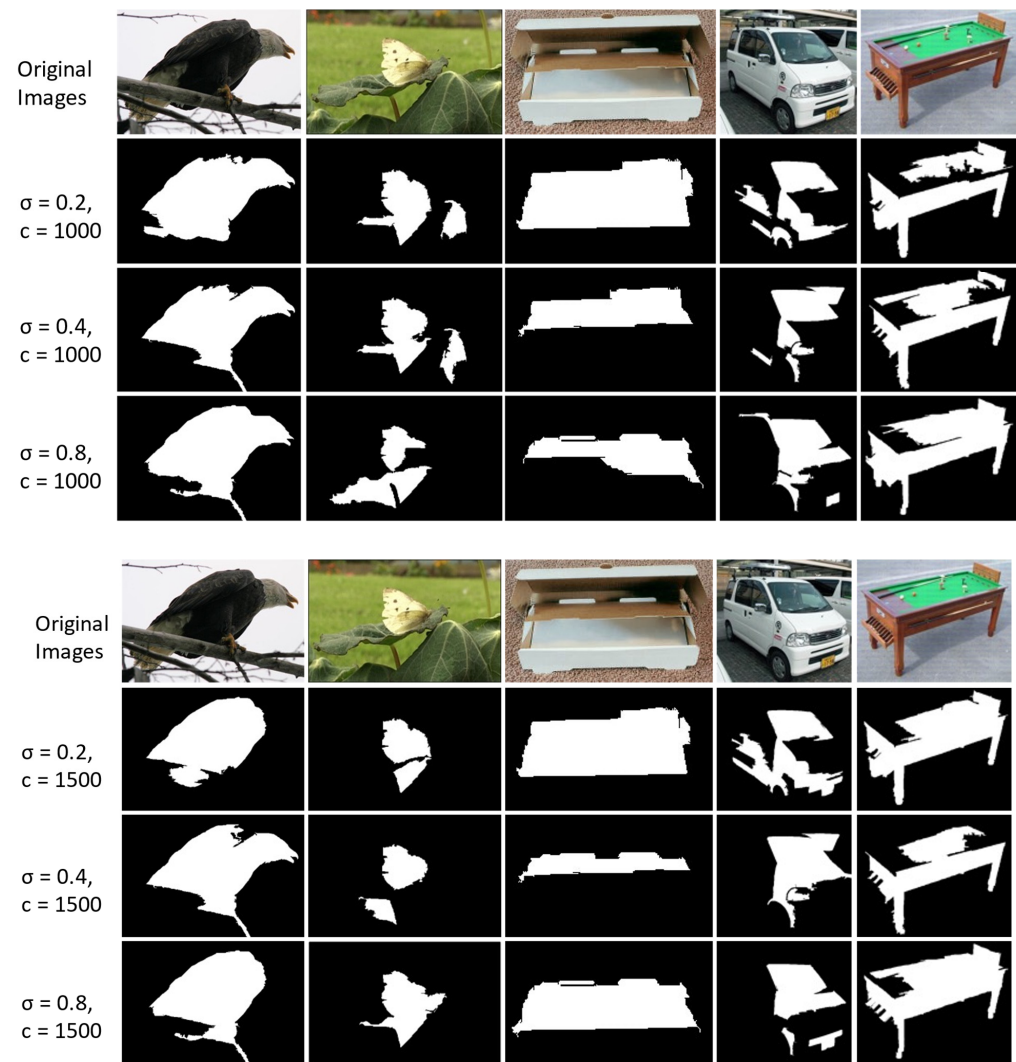


Figure 6. The heuristic binary masks are generated using DRFI with $\sigma = \{0.2, 0.4, 0.8\}$ with $c = \{1000, 1500\}$.

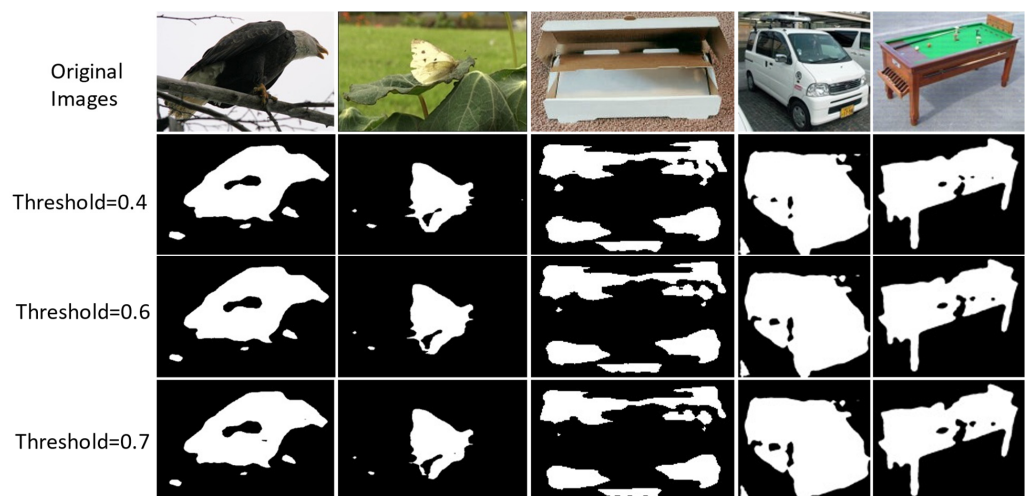


Figure 7. The heuristic binary masks are generated using an unsupervised deep learning encoder with saliency threshold values.

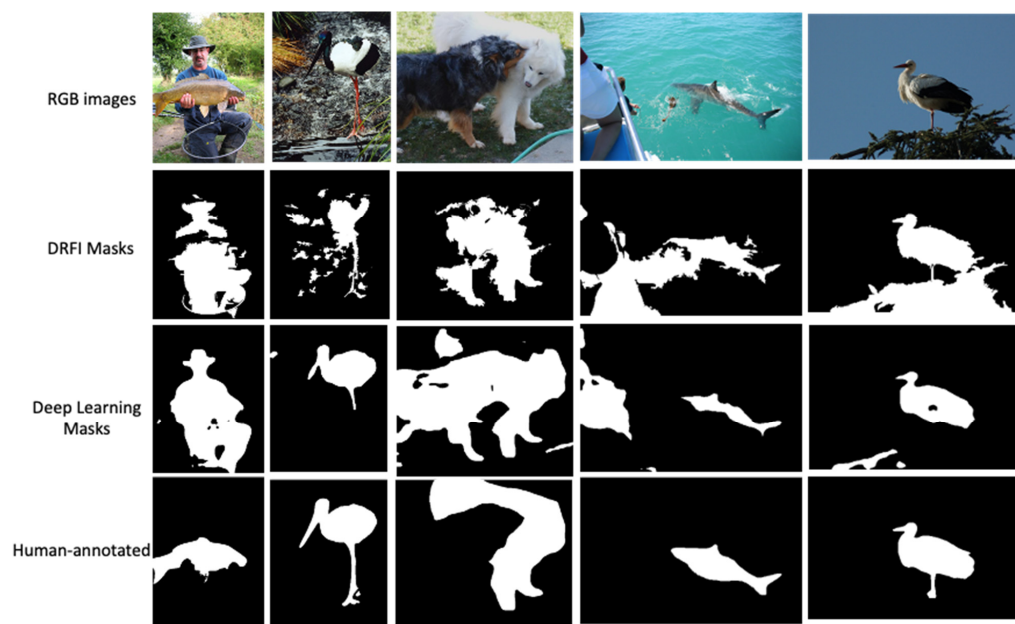


Figure 8. The inspection examples of the generated heuristic binary masks between DRFI and deep learning.

Table 5. The impact of mask quality on HARL framework performance on the downstream object detection and instance segmentation task. We use Faster R-CNN with C4 backbone for object detection and Mask-RCNN with FPN backbone for instance segmentation. Object detection and instance segmentation backbones are initialized with the 100-epoch pre-trained ResNet-50 backbone on ImageNet dataset. The best result is bolded.

Method	Object Detection						Instance Segmentation		
	VOC07 + 12 Detection			COCO Detection			COCO Segmentation		
	AP ₅₀	AP	AP ₇₅	AP ₅₀	AP	AP ₇₅	AP ₅₀ ^{mask}	AP ^{mask}	AP ₇₅ ^{mask}
HARL (DRFI Masks)	82.3	55.4	61.2	44.2	24.6	24.8	41.8	24.3	25.1
HARL (Deep Learning Masks)	82.1	55.5	61.7	44.7	24.7	25.3	42.3	24.6	25.2

6. Conclusions and Future Work

We introduce the HARL framework, a new self-supervised visual representation learning framework, by leveraging visual attention with the heuristic binary mask. As a result, HARL manages higher-quality semantical information that considerably improves representation learning of self-supervised pretraining compared to previous state-of-the-art methods [9,13,17,18] on semi-supervised and transfers learning on various benchmarks. The two main advantages of the proposed method include: (i) the early attention mechanism that can be applied across different natural image datasets because we use unsupervised techniques to generate the heuristic mask and do not rely on external supervision; (ii) the entire framework can transfer and adapt quickly either to self-supervised contrastive or non-contrastive learning framework. Furthermore, our method will apply and accelerate the currently self-supervised learning direction on pixel-level objectives. Our object-level abstract will make this objective more efficient than the existing work based on computing pixel distance [61].

In our HARL framework, the heuristic binary mask is critical. However, the remaining challenge of estimating accurate masks is suitable for datasets with one primary object, such as the ImageNet dataset. The alternative is mining the object proposal of the image in the complex dataset which contains multiple things by producing heuristic semantic segmentation masks. Designing the new self-supervised framework to solve the remaining

challenge of datasets which contain multiple objects is an essential next step and exciting research direction for our future work.

Supplementary Materials: The following documents support our experimental results reported in this study. Our code implementation on PyTorch implementation (https://github.com/TranNhiem/Heuristic_Attention_Representation_Learning_SSL_Pytorch accessed on 19 September 2021) and TensorFlow (https://github.com/TranNhiem/Heuristic_Attention_Representation_Learning_SSL_Tensorflow accessed on 26 November 2021). Our experimental results and report included in different sections can be downloaded at: <https://www.hh-ri.com/2022/05/30/heuristic-attention-representation-learning-for-self-supervised-pretraining/> (accessed on 30 May 2021).

Author Contributions: Conceptualization, Y.-H.L.; methodology, Y.-H.L. and V.N.T.; software, S.-H.L. and V.N.T.; validation, S.-H.L. and V.N.T.; writing—original draft preparation, Y.-H.L. and V.N.T.; writing—review and editing, Y.-H.L. and V.N.T.; supervision, Y.-H.L. and J.-C.W.; project administration, Y.-H.L. and J.-C.W.; funding acquisition, Y.-H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: In this study, we construct two novel sets of heuristic binary mask datasets for the ImageNet ILSVRC training set, which can be found here: <https://www.hh-ri.com/2022/05/30/heuristic-attention-representation-learning-for-self-supervised-pretraining/> (accessed on 30 May 2022).

Acknowledgments: The authors would like to thank the following people for their help throughout the process of building this study: Timothy Ko for helping create the heuristic binary mask on the DRFI technique for the ImageNet ILSVRC-2012 dataset and Kai-Lin Yang who provided the setting virtual machines for experiments and gave suggestions during the development of the project.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Implementation Detail

Appendix A.1. Implementation Data Augmentation

HARL data augmentation pipeline starts with the standard inception-style random cropping [62]. These cropping views continue to transform using the same set of image augmentation as in SimCLR [9], consisting of the arbitrary sequence composition transformation (color distortion, grayscale conversion, gaussian blur, solarization).

Each RGB image and the heuristic binary mask corresponding to each image are transformed through the augmentation pipeline composed of the following operations described below. First, we utilize the image with the random crop with resizing and random flipping. For the binary mask, these masks apply, only cropping and flipping the underlying RGB image which corresponds. Then, these crop images used give the probability of color distortion (color jittering, color dropping), random Gaussian blur and solarization.

1. Random cropping with resizes: a random patch of the image is selected. In our pipeline, we use the inception-style random cropping [62], whose area crop is uniformly sampled in [0.08 to 1.0] of the original image, and the random aspect ratio is logarithmically sampled in [3/4, 4/3]. The patch is then resized to 224×224 pixels using bicubic interpolation;
2. Optional horizontal flipping (left and right);
3. Color jittering: the brightness, contrast, saturation and hue are shifted by a uniformly distributed offset;
4. Optional color dropping: the RGB image is replaced by its greyscale values;
5. Gaussian blurring with a 224×224 square kernel and a standard deviation uniformly sampled from [0.1, 2.0];
6. Optional solarization: a point-wise color transformation $x \mapsto x \cdot \mathbb{1}_{x < 0.5} + (1 - x) \cdot \mathbb{1}_{x \geq 0.5}$ for pixel values in the range [0–1].

The two views augmented image's x, x' and mask pair m, m' results from augmentations sample from distributions T, T', M and M' , respectively. These distributions apply the primitives described above with different probabilities and magnitudes shown in Table A1. The following table specifies these parameters' inheritance from the BYOL framework [13] without modification.

Table A1. Parameters used to generate image augmentations.

Parameter	T	T'	M	M'
Inception-style random crop probability	1.0	1.0	1.0	1.0
Flip probability	0.5	0.5	0.5	0.5
Color jittering probability	0.8	0.8	-	-
Brightness adjustment max intensity	0.4	0.4	-	-
Contrast adjustment max intensity	0.4	0.4	-	-
Saturation adjustment max intensity	0.2	0.2	-	-
Hue adjustment max intensity	0.1	0.1	-	-
Color dropping probability	0.2	0.2	-	-
Gaussian blurring probability	1.0	0.1	-	-
Solarization probability	0.0	0.2	-	-

Appendix A.2. Implementation Masking Feature

The masking feature step of the HARL framework is essential to leverage the objective-level information from the heuristic binary mask. The masking features method is composed of three steps. The first step is taking the spatial feature map output $7 \times 7 \times 2048$, which is the final layer before the global average output pooling of the ResNet architecture. Second, in our training loop design, the mask image is directly resized to $7 \times 7 \times 3$ to match the size of the output spatial feature maps without passing through the encoder. Then, the resized mask indexes the feature, one encodes for the foreground feature and zero encodes for the background feature. In the end, we multiply the indexing mask with the spatial features maps to separate the foreground and background features (the corresponding output is two spatial features maps of $7 \times 7 \times 2048$ for foreground and background features). Then, these two spatial feature map outputs apply global average pooling and further reduce dimension with non-linear multi-layer perceptron (MLP) architecture.

Appendix B. Evaluation on the ImageNet and Transfer Learning

Appendix B.1. Implementation Masking Feature Linear Evaluation Semi-Supervised Protocol on ImageNet

Our data preprocessing procedure is described as follows: At training time, the images apply the simple augmentations strategies, including random flip and crops with resizing to 224×224 pixels. At testing time, all images applied are resized to 256 pixels along the shorter side using bicubic resampling, which took a 224×224 center crop. Images are normalized by color channel in training time and testing and divided by standard deviation computed on ImageNet ([9,13] provide a similar pipeline for data processing).

Linear evaluation: We train a linear classifier on top of the frozen pre-trained encoder representation in the linear evaluation without updating the network parameters and the batch statistics. In design and configuration protocol, we follow the standard on ImageNet as in [9,51,54,55]. To train and optimize the linear classifier, we use the SGD optimizer to optimize the cross-entropy loss with the Nesterov momentum over 100 epochs using a batch size of 1024 and a momentum of 0.9. without regularization methods such as weight decay, gradient clipping [63], etc. We report the test set's accuracy (the public validation set of the original ILSVRC2012 ImageNet [24] dataset).

Semi-supervised evaluation: We fine-tuned the network parameters of the pre-trained encoder representation following the semi-supervised learning protocol and procedure as in [9]. Data preprocessing and augmentation strategies at training and testing time for 1% and 10% follow a similar procedure of linear evaluation (described in Appendix C.1) except that with a larger batch size of 2048 and trained over 60 epochs for 1% labeled data

and 30 epochs for 10% labeled data. In Table 1 of Section 5.1, we report that the result fine-tuned the representation over the 1% and 10% ImageNet splits from [9] with ResNet-50 ($1\times$) architectures.

Datasets: We followed previous works [9,13] to transfer the representation on the linear classification and fine-tuned it on six different natural image datasets. These datasets are namely Food-101 [64], CIFAR-10 [65] and CIFAR-100 [65], the SUN397 scene dataset [66], Stanford Cars [67] and the Describable Textures Dataset (DTD) [68]. The detail of each dataset is described in Table A2. We use the training set and validation set, which are specified by the dataset creators, to select hyperparameters. On datasets without a test set or validation set, we use the validation examples as a test set or hold out a subset of the training examples we use as the validation set, as described in Table A2.

Table A2. The different image datasets used in transfer learning. When an official test split with labels is not publicly available, we use the official validation split as a test set and create a held-out validation set from the training examples.

Dataset	Classes	Original Training Examples	Training Examples	Validation Examples	Test Examples	Accuracy Measure	Test Provided
Food101	101	75,750	68,175	7575	25,250	Top-1 accuracy	-
CIFAR-10	10	50,000	45,000	5000	10,000	Top-1 accuracy	-
CIFAR-100	100	50,000	44,933	5067	10,000	Top-1 accuracy	-
Sun397 (split 1)	397	19,850	15,880	3970	19,850	Top-1 accuracy	-
Cars	196	8144	6494	1650	8041	Top-1 accuracy	-
DTD (split 1)	47	1880	1880	1880	1880	Top-1 accuracy	Yes

Standard evaluation metrics: To evaluate HARL transfer learning on different datasets and other vision tasks, we use the standard evaluation metrics of each dataset to assess and benchmark our results on these datasets as described in Top-1, AP, AP₅₀ and AP₇₅.

- Top-1: We compute the proportion of correctly classified examples.
- AP, AP₅₀ and AP₇₅: We compute the average precision as defined in [56].

Appendix B.2. Transfer via Linear Classification and Fine-Tuning

Transfer linear classification: We initialize the network parameters and freeze the pre-trained encoder without updating the network parameters and batch statistics. The standard linear evaluation protocol follows [9,51,55]. In training and testing, the images are resized to 224×224 along the shorter side using bicubic resampling and then normalized with ImageNet statistics without data augmentation. Both phase images normalized the color channels by subtracting the average color and dividing by the standard deviation. We train a regularized multinomial logistic regression classifier on top of the frozen representation. We optimize cross-entropy loss ℓ_2 —regularization with the parameters from a range of 45 logarithmically spaced values between 10^{-6} and 10^5 (similar to the optimization procedure of [13]). The model is retrained on the training and validation set combined. The model accuracy performance is reported for the test set.

Transfer fine-tuning: We follow fine-tuning protocol as in [9,51,69] to initialize the network with the parameters of the pre-trained representation. At both phase training and testing time, we follow the image preprocessing and data augmentation strategies to the linear evaluation procedure in Appendix B.1. To fine-tune the network, we optimized the cross-entropy loss using SGD optimizer with a Nesterov momentum value of 0.9 and trained over 20,000 steps with a batch size of 256. We set a hyperparameter including the momentum parameter for batch statistics, learning rate and weight decay selection method, same as in [9,13]. After selecting the optimal hyperparameters configured for the validation set, the model is retrained on the combined training and validation set together, using the specified parameters. The absolute accuracy is reported for the test set.

Appendix B.3. Transfer Learning to Other Vision Tasks

Object detection and instance segmentation: We followed previous works [13,17] for the standard setup transferring procedure on Pascal object detection. We use a Faster R-CNN [57] with the R50-C4 backbone. We fine-tune with the training and validation set (16K images) and report the results for the test set of the PASCAL VOC07 + 12 [56] dataset. The backbone is initialized with our pre-trained ResNet50. We use the SGD optimizer to optimize network parameters for 24K iterations with a batch size of 16. We use the initial learning rate of 0.08, then it is reduced to 10^{-2} at 18K and 10^{-3} at 22K with a linear warmup of the slope 0.3333 for 1000 iterations and the region proposal loss weight of 0.2. Then, we report the final results of AP, AP₅₀ and AP₇₅ metrics for the test set. For instance, regarding the segmentation task on the COCO [58] dataset, we use Mask R-CNN with FPN backbone to iterate over 90K iterations with a batch size of 16. We initialize the learning rate at 0.03 and reduce it by 10 at the 60K and 80K iterations with warmup iterations of 50.

Appendix C. Heuristic Mask Proposal Methods

In our HARL framework, to generate the heuristic binary mask we investigated various supervised and unsupervised techniques from conventional machine learning to deep-learning-based approaches. The benchmark qualitative and quantitatively state-of-the-art approaches use computer vision methods [70]. The comprehensive literature survey and benchmark [71] offer multiple supervised deep-learning-based methods for salient object detection on multi-level supervision, network architectures and learning paradigms. Several works of the unsupervised deep learning method [72,73] used predictions obtained with the hand-crafted prior as the pseudo label to train the deep neural network.

Appendix C.1. Heuristic Binary Mask Generates Using DRFI

Our first approach uses the conventional machine learning method to generate binary masks by adopting the DRFI [44] technique. This method detects a salient object inside an image by carrying out three main steps: multi-level segmentation that segments an image into regions and regional saliency computation that maps the features extracted from each area to a saliency score, which is predicted by a random forest based on three elements: regional contrast, regional property and regional backgrounds. Additionally, at last, multi-level saliency fusion combines the saliency maps over all the layers of segmentation to obtain the final saliency map. To obtain a binary mask, we generate the saliency map of an image. Then, we define a threshold of 40% (top 40% saliency score) to determine what regions are considered salient objects. Any area that is not a salient object will be regarded as background. We generate a diverse set of binary masks by varying the two hyperparameters σ and the minimum cluster size c . Using $\sigma \in \{0.2, 0.4, 0.8\}$ and $c \in \{1000, 1500\}$ in our implementation, we defined $\sigma = 0.8$ and $c = 1000$ for generating masks in the ImageNet dataset. Additionally, the different configuration hyperparameters experimented with sweeping sigma values $\sigma = \{0.2, 0.4, 0.8\}$ and component sizes of $c = \{1000, 1500\}$ are shown in Figure 6.

Appendix C.2. Heuristic Binary Mask Generates Using Unsupervised Deep Learning

The second approach in our mask-generated techniques is based on a self-supervised pre-trained feature extractor from previous works [9,17,39,42]. We design a new saliency head prediction with pre-trained encoder representation to generate the binary masks. The design is to obtain a binary mask by carrying out three main steps. First, we take the output feature maps from a pre-trained ResNet-50 encoder [9,42]. Second, we pass the output feature map into a 1×1 convolutional classification layer for saliency prediction. The classification layer predicts the saliency or “foregroundness” of a pixel. Finally, we take the classification layer’s output values and set a threshold to decide which pixels belong to the foreground. The pixel saliency value more significant than the threshold is determined as a foreground object. In our implementation, we defined a threshold value equal to 0.5 for

generating masks in the ImageNet dataset. We further experiment with several threshold values in {0.4, 0.6, 0.7}; all these configure mask-generated examples in Figure 7.

References

1. Shu, Y.; Kou, Z.; Cao, Z.; Wang, J.; Long, M. Zoo-tuning: Adaptive transfer from a zoo of models. *arXiv* **2021**, arXiv:2106.15434.
2. Yang, Q.; Zhang, Y.; Dai, W.; Pan, S.J. *Transfer Learning*; Cambridge University Press: Cambridge, UK, 2020.
3. You, K.; Kou, Z.; Long, M.; Wang, J. Co-Tuning for Transfer Learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 17236–17246.
4. Misra, I.; Shrivastava, A.; Gupta, A.; Hebert, M. Cross-Stitch Networks for Multi-task Learning. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 3994–4003.
5. Li, X.; Xiong, H.; Xu, C.; Dou, D. SMILE: Self-distilled mixup for efficient transfer learning. *arXiv* **2021**, arXiv:2103.13941.
6. Tishby, N.; Zaslavsky, N. Deep learning and the information bottleneck principle. In Proceedings of the 2015 IEEE Information Theory Workshop (ITW), Jeju Island, Korea, 11–15 October 2015; pp. 1–5.
7. Shwartz-Ziv, R.; Tishby, N. Opening the black box of deep neural networks via information. *arXiv* **2017**, arXiv:1703.00810.
8. Amjad, R.A.; Geiger, B.C. Learning representations for neural network-based classification using the information bottleneck principle. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2225–2239. [[CrossRef](#)]
9. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G.E. A simple framework for contrastive learning of visual representations. *arXiv* **2020**, arXiv:2002.05709.
10. Goyal, P.; Caron, M.; Lefaudeaux, B.; Xu, M.; Wang, P.; Pai, V.; Singh, M.; Liptchinsky, V.; Misra, I.; Joulin, A.; et al. Self-supervised Pretraining of Visual Features in the Wild. *arXiv* **2021**, arXiv:2103.01988.
11. Misra, I.; Maaten, L.v.d. Self-supervised learning of pretext-invariant representations. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 6706–6716.
12. Ermolov, A.; Siarohin, A.; Sangineto, E.; Sebe, N. Whitening for self-supervised representation learning. In Proceedings of the International Conference on Machine Learning ICML, Virtual, 18–24 July 2021.
13. Grill, J.-B.; Strub, F.; Altch'e, F.; Tallec, C.; Richemond, P.H.; Buchatskaya, E.; Doersch, C.; Pires, B.v.; Guo, Z.D.; Azar, M.G.; et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv* **2020**, arXiv:2006.07733.
14. Caron, M.; Touvron, H.; Misra, I.; J'egou, H.e.; Mairal, J.; Bojanowski, P.; Joulin, A. Emerging properties in self-supervised vision transformers. *arXiv* **2021**, arXiv:2104.14294.
15. Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv* **2020**, arXiv:2006.09882.
16. Bromley, J.; Bentz, J.W.; Bottou, L.; Guyon, I.; LeCun, Y.; Moore, C.; Säckinger, E.; Shah, R. Signature verification using a “Siamese” time delay neural network. In Proceedings of the 13th International Joint Conference on Artificial Intelligence, Chambéry, France, 28 August–3 September 1993.
17. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R.B. Momentum contrast for unsupervised visual representation learning. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 9726–9735.
18. Chen, X.; He, K. Exploring simple siamese representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition CVPR, Nashville, TN, USA, 20–25 June 2021.
19. Hayhoe, M.M.; Ballard, D.H. Eye movements in natural behavior. *Trends Cogn. Sci.* **2005**, *9*, 188–194. [[CrossRef](#)] [[PubMed](#)]
20. Borji, A.; SihiteDicky, N.; Itti, L. Quantitative analysis of human-model agreement in visual saliency modeling. *IEEE Trans. Image Process.* **2013**, *22*, 55–69. [[CrossRef](#)] [[PubMed](#)]
21. Benois-Pineau, J.; Callet, P.L. Visual content indexing and retrieval with psycho-visual models. In *Multimedia Systems and Applications*; Springer: Cham, Switzerland, 2017.
22. Awh, E.; Armstrong, K.M.; Moore, T. Visual and oculomotor selection: Links, causes and implications for spatial attention. *Trends Cogn. Sci.* **2006**, *10*, 124–130. [[CrossRef](#)] [[PubMed](#)]
23. Tian, Y.; Chen, X.; Ganguli, S. Understanding self-supervised learning dynamics without contrastive Pairs. *arXiv* **2021**, arXiv:2102.06810.
24. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.S.; et al. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [[CrossRef](#)]
25. Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th International Conference on Machine Learning ICML, Helsinki, Finland, 5–9 July 2008.
26. Bojanowski, P.; Joulin, A. Unsupervised learning by predicting noise. *arXiv* **2017**, arXiv:1704.05310.
27. Larsson, G.; Maire, M.; Shakhnarovich, G. Colorization as a proxy task for visual understanding. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 840–849.
28. Iizuka, S.; Simo-Serra, E. Let there be color!: Joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Trans. Graph. (ToG)* **2016**, *35*, 1–11. [[CrossRef](#)]
29. Pathak, D.; Krähenbühl, P.; Donahue, J.; Darrell, T.; Efros, A.A. Context encoders: Feature learning by inpainting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2536–2544.
30. Gidaris, S.; Singh, P.; Komodakis, N. Unsupervised representation learning by predicting image rotations. *arXiv* **2018**, arXiv:1803.07728.

31. Noroozi, M.; Favaro, P. Unsupervised learning of visual representations by solving jigsaw puzzles. In Proceedings of the European Conference on Computer Vision ECCV, Amsterdam, The Netherlands, 8–16 October 2016.
32. Zhang, R.; Isola, P.; Efros, A.A. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 645–654.
33. Mundhenk, T.N.; Ho, D.; Chen, B.Y. Improvements to context based self-supervised learning. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 9339–9348.
34. Donahue, J.; Krähenbühl, P.; Darrell, T. Adversarial feature learning. *arXiv* **2017**, arXiv:1605.09782.
35. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.C.; Bengio, Y. Generative adversarial nets. In Proceedings of the Neural Information Processing Systems NIPS, Montreal, QC, Canada, 8–13 December 2014.
36. Donahue, J.; Simonyan, K. Large scale adversarial representation learning. In Proceedings of the Neural Information Processing Systems NeurIPS, Vancouver, BC, Canada, 8–14 December 2019.
37. Bansal, V.; Buckchash, H.; Raman, B. Discriminative auto-encoding for classification and representation learning problems. *IEEE Signal Process. Lett.* **2021**, *28*, 987–991. [[CrossRef](#)]
38. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.
39. Chen, T.; Kornblith, S.; Swersky, K.; Norouzi, M.; Hinton, G.E. Big Self-supervised models are strong semi-supervised learners. *arXiv* **2020**, arXiv:2006.10029.
40. Jaiswal, A.; Babu, A.R.; Zadeh, M.Z.; Banerjee, D.; Makedon, F. A Survey on contrastive self-supervised learning. *Technologies* **2020**, *9*, 2. [[CrossRef](#)]
41. Hinton, G.E.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.
42. Van Gansbeke, W.; Vandenhende, S.; Georgoulis, S.; Gool, L.V. unsupervised semantic segmentation by contrasting object mask proposals. *arXiv* **2021**, arXiv:2102.06191.
43. Zhang, X.; Maire, M. Self-Supervised visual representation learning from hierarchical grouping. *arXiv* **2020**, arXiv:2012.03044.
44. Jiang, H.; Yuan, Z.; Cheng, M.-M.; Gong, Y.; Zheng, N.; Wang, J. Salient object detection: A discriminative regional feature integration approach. *Int. J. Comput. Vis.* **2013**, *123*, 251–268.
45. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
46. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arXiv:1502.03167.
47. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning ICML, Haifa, Israel, 21–24 June 2010.
48. You, Y.; Gitman, I.; Ginsburg, B. Scaling SGD batch size to 32K for imageNet training. *arXiv* **2017**, arXiv:1708.03888.
49. Loshchilov, I.; Hutter, F. SGDR: Stochastic gradient descent with warm restarts. *arXiv* **2017**, arXiv:1608.03983.
50. Goyal, P.; Dollár, P.; Girshick, R.B.; Noordhuis, P.; Wesolowski, L.; Kyrola, A.; Tulloch, A.; Jia, Y.; He, K. Accurate, large Minibatch SGD: Training ImageNet in 1 hour. *arXiv* **2017**, arXiv:1706.02677.
51. Kolesnikov, A.; Zhai, X.; Beyer, L. Revisiting self-supervised visual representation learning. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 1920–1929.
52. Ye, M.; Zhang, X.; Yuen, P.; Chang, S.-F. Unsupervised embedding learning via invariant and spreading instance feature. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 6203–6212.
53. Hjelm, R.D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Trischler, A.; Bengio, Y. Learning deep representations by mutual information estimation and maximization. *arXiv* **2019**, arXiv:1808.06670.
54. Chen, X.; Fan, H.; Girshick, R.B.; He, K. Improved baselines with momentum contrastive learning. *arXiv* **2020**, arXiv:2003.04297.
55. Kornblith, S.; Shlens, J.; Le, Q.V. Do Better ImageNet models transfer better? In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 2656–2666.
56. Everingham, M.; Gool, L.V.; Williams, C.K.I.; Winn, J.M.; Zisserman, A. The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vis.* **2009**, *88*, 303–338. [[CrossRef](#)]
57. Ren, S.; He, K.; Girshick, R.B.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *39*, 1137–1149. [[CrossRef](#)]
58. Lin, T.-Y.; Maire, M.; Belongie, S.J.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. In Proceedings of the European Conference on Computer Vision ECCV, Zurich, Switzerland, 6–12 September 2014.
59. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R.B. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 386–397. [[CrossRef](#)]
60. Zhang, S.; Liew, J.H.; Wei, Y.; Wei, S.; Zhao, Y. Interactive object segmentation with inside-outside guidance. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 12231–12241.
61. Xie, Z.; Lin, Y.; Zhang, Z.; Cao, Y.; Lin, S.; Hu, H. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 16679–16688.

62. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.E.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
63. Pascanu, R.; Mikolov, T.; Bengio, Y. On the difficulty of training recurrent neural networks. In Proceedings of the International Conference on Machine Learning ICML, Atlanta, GA, USA, 16–21 June 2013.
64. Bossard, L.; Guillaumin, M.; Gool, L.V. Food-101-mining discriminative components with random forests. In Proceedings of the European Conference on Computer Vision ECCV, Zurich, Switzerland, 6–12 September 2014.
65. Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images. Master's Thesis, University of Toronto, Toronto, ON, Canada, 2009. Available online: <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf> (accessed on 8 April 2009).
66. Xiao, J.; Hays, J.; Ehinger, K.A.; Oliva, A.; Torralba, A. SUN database: Large-scale scene recognition from abbey to zoo. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 3485–3492.
67. Krause, J.; Stark, M.; Deng, J.; Fei-Fei, L. 3D Object Representations for fine-grained categorization. In Proceedings of the 2013 IEEE International Conference on Computer Vision Workshops, Sydney, Australia, 2–8 December 2013; pp. 554–561.
68. Cimpoi, M.; Maji, S.; Kokkinos, I.; Mohamed, S.; Vedaldi, A. Describing textures in the wild. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 3606–3613.
69. Hénaff, O.J.; Srinivas, A.; Fauw, J.D.; Razavi, A.; Doersch, C.; Eslami, S.M.A.; Oord, A.R.V.D. Data-efficient image recognition with contrastive predictive coding. *arXiv* **2020**, arXiv:1905.09272.
70. Borji, A.; Cheng, M.-M.; Jiang, H.; Li, J. Salient object detection: A benchmark. *IEEE Trans. Image Process.* **2015**, *24*, 5706–5722. [[CrossRef](#)]
71. Wang, W.; Lai, Q.; Fu, H.; Shen, J.; Ling, H. Salient object detection in the deep learning era: An in-depth survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 3239–3259. [[CrossRef](#)]
72. Zou, W.; Komodakis, N. HARK: Hierarchy-associated rich features for salient object detection. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 406–414.
73. Zhang, J.; Zhang, T.; Dai, Y.; Harandi, M.; Hartley, R.I. Deep unsupervised saliency detection: A multiple noisy labeling perspective. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 9029–9038.