Exploring Localization for Self-supervised Fine-grained Contrastive Learning

Di Wu\textsuperscript{1,2,⋆}
wudi@westlake.edu.cn
Siyuan Li\textsuperscript{1,2,⋆}
lisiyuan@westlake.edu.cn
Zelin Zang\textsuperscript{1,2}
zangzelin@westlake.edu.cn
Stan Z. Li\textsuperscript{1,†}
Stan.ZQ.Li@westlake.edu.cn

\textsuperscript{1} AI Lab, School of Engineering, Westlake University, Hangzhou, Zhejiang, China
\textsuperscript{2} Zhejiang University, Hangzhou, Zhejiang, China

Abstract

Self-supervised contrastive learning has demonstrated great potential in learning visual representations. Despite their success in various downstream tasks such as image classification and object detection, self-supervised pre-training for fine-grained scenarios is not fully explored. We point out that current contrastive methods are prone to memorizing background/foreground texture and therefore have a limitation in localizing the foreground object. Analysis suggests that learning to extract discriminative texture information and localization are equally crucial for fine-grained self-supervised pre-training. Based on our findings, we introduce cross-view saliency alignment (CVSA), a contrastive learning framework that first crops and swaps saliency regions of images as a novel view generation and then guides the model to localize on foreground objects via a cross-view alignment loss. Extensive experiments on both small- and large-scale fine-grained classification benchmarks show that CVSA significantly improves the learned representation.

1 Introduction

Learning visual representations without supervision by leveraging pretext tasks has become increasingly popular. Various learning approaches such as colorization [56], Rel-Loc [31], Rot-Pred [13] have been proposed to learn such representations. The objective of these pretext tasks is to capture invariant features through predicting transformations applied to the same image. More recently, self-supervised representation learning has witnessed significant progress by the use of contrastive loss [4, 16, 17, 18, 28]. Despite that contrastive-based methods have even outperformed supervised methods under some circumstances, their success has largely been confined to large-scale general-purpose datasets (coarse-grained) such as ImageNet [22]. We argue that current contrastive learning methods only work on coarse-grained iconic images with large foreground objects residing in the background with infor-
Figure 1: Comparison of RandomResizedCrop (RRC) and the proposed Saliency Swap (SS). We visualize Grad-CAM [35] of linear classifiers for pre-trained models. (a) shows the most commonly adopted RRC in contrastive learning (CL) methods. The random nature may cause views to contain mainly the backgrounds of the image leading to semantic inconsistency across views. (b) shows our proposed SS, which crops from regions of interest of the reference image and replaces the saliency regions of two randomly selected background images to guarantee semantic consistency.

mative discriminative texture (e.g., ImageNet) but perform poorly when background texture provides little clue (e.g., CUB-200-2011 [42]) for fine-grained separation.

To bridge the substantial gap between self-supervised and supervised representation learning on fine-grained object recognition, we first analyze and compare knowledge learned by various self-supervised methods and supervised methods during pre-training. We find that current self-supervised contrastive learning methods tend to learn low-level texture information and lack the localization ability of the foreground object. In contrast, the supervised method shows better localization ability. Specifically, we show that the incompetence of localization of current contrastive learning is primarily due to the commonly adopted RandomResizedCrop (RRC) augmentation, where a random size patch at a random location is cropped and resized to the original size. The model then might learn a semantic representation of the bird by contrasting the tree and the wing of the bird, as illustrated in Figure 1. This practice may be reasonable for coarse-grained recognition if background cues are more associated with the class than the foreground cues (e.g. $p(\text{bird}|\text{tree}) > p(\text{car}|\text{tree})$). However, the background of the image being a tree is not as informative when distinguishing bird species. Consequently, the model learns by cheating on picking low-level texture clues (usually from the background) instead of learning by localizing the foreground. This phenomenon is mutual for existing contrastive methods such as MoCo.v2[5], BYOL[15] despite different contrastive mechanisms.

The devil lies in semantically discriminative fine-grained feature extraction for a successful contrastive pre-training. To remedy the inadequacy of fine-grained feature capturing due to failure in localizing to discriminative regions, we propose to empower contrastive learning with localization ability by aligning fine-grained semantic features across augmented views. In particular, we come up with a pre-training framework called Cross-View Saliency Alignment (CVSA). CVSA consists of two algorithmic components: (a) A general plug-and-play data augmentation strategy called SaliencySwap, which swaps the saliency region of the reference image with the saliency region of a randomly selected background image. SaliencySwap ensures semantic consistency between augmented views while introducing
background variation. A demonstration of SaliencySwap in comparison with RRC is shown in Figure 1. An alignment loss that provides an explicit localization supervision signal by forcing the model to give the highest correspondence response intensity of the foreground object across views.

On top of the proposed CVSA, to further bridge the performance gap between self-supervised and supervised representation learning on the fine-grained recognition problem, we offer a dual-stage pre-training setting, which utilizes coarse-grained datasets for low-level feature extraction and fine-grained datasets for high-level target discrimination and localization. In short, this paper makes the following contributions:

• We delve deep into the knowledge learned by various self-supervised methods compared to supervised methods during the fine-grained pre-training phase and point out the cause of limitations.
• We develop a novel contrastive learning framework for fine-grained recognition, which contains a data augmentation technique called SaliencySwap to guarantee semantic consistency between views and an alignment objective which enables the model to localize.
• Extensive experiments show consistent performance gain of CVSA under various pre-training stage settings on small- and large-scale fine-grained benchmarks.

2 Cross-view Saliency Alignment

We explore the capabilities learned out of three classes of pre-training mechanisms, namely self-supervised contrastive, non-contrastive, and supervised methods. In particular, we focus on discriminative feature extraction and object localization ability. Without loss of generality, we select MoCo.v2, BYOL, Rot-Pred, and supervised classification for comparison. We discover that compared to supervised methods, self-supervised methods show worse object localization ability and discriminative feature extraction ability is also crucial for fine-grained categorization. The details of the experiments and corresponding analysis are provided in Appendix A. Based on our observations, given a fine-grained classification problem, similar to [1], we assume $X$ to be a set of all samples with an underlying set of discrete latent classes $C$ that represent semantic content, we obtain the joint distribution between each sample $x$ and its class $c$:

$$p(c, x) = p(c|x_{\text{fore}}) \cdot p(x_{\text{fore}}|x),$$  \hspace{1cm} (1)$$

where $x_{\text{fore}}$ stands for the foreground object. The intuition behind this factorization suggests that given an image of a fine-grained object, the model should localize the foreground object ($p(x_{\text{fore}}|x)$) to discriminate the species of the foreground object ($p(c|x_{\text{fore}})$). Following this formulation, we propose a dual-stage pre-training pipeline for self-supervised fine-grained recognition with the first-stage learning discriminative texture extraction ability and second-stage learning localization capability.

2.1 SaliencySwap

SaliencySwap maximally utilizes the saliency information for foreground semantic consistency across views while introducing background variation. SaliencySwap guarantees that each view at least contains part of the foreground object and thus prevents the encoder from learning irrelevant feature representation through pure background information.
2.1.1 Source Saliency Detection

A saliency detection algorithm generates a saliency map that indicates the objects of interest (primarily foreground). Let $I \in \mathbb{R}^{W \times H \times C}$ be an image in the training set, define $\psi$ to be a saliency detection algorithm, then the output saliency map $S_{i,j} = \psi(I_{i,j}) \in \mathbb{R}^{W \times H}$ indicates the saliency intensity value at pixel $I_{i,j}$. The saliency information can be noisy. Therefore, we seek to find a bounding box $B = (l,t, W_b, H_b)$ of the foreground object with the highest averaged saliency information satisfying the following objective function:

$$\text{argmax}_{W_b,H_b,l,t} \sum_{i=l}^{i=l+W_b} \sum_{j=t}^{j=t+H_b} S_{i,j} \frac{1}{W_b \times H_b}.$$  \hspace{1cm} (2)

A corresponding binary saliency mask $M \in \mathbb{R}^{W \times H}$ is defined by filling with 1 within the bounding box $B$, otherwise 0. Then we crop a random patch within the bounding box $B$. Similar to RRC, the size of the patch is determined based on an area ratio (to the area of the bounding box), which is sampled from a uniform distribution $U(\lambda, 1)$.

2.1.2 Foreground Background Fusion

We then combine the cropped foreground patch from the source image (foreground image) with another randomly selected image (background image). To avoid saliency ambiguity, we restrict each augmented view to contain the saliency information only of one semantic object. We consider two ways of merging: (I) The background dataset is the same as the foreground dataset. The saliency information of the background needs to be eliminated. We first calculate the bounding box $B_f, B_b$ of the foreground and background images, respectively, using Eqn. 2. For scenic images, we use the bounding box with maximum area. Then select a random patch from the foreground $B_f$ and resize it to the shape of $B_b$. Finally, we replace $B_b$ with the resized foreground patch. (II) The background dataset is different from the foreground dataset. We choose a dataset like COCO rather than the iconic dataset like IN such that the background dataset is rich in environments. Again, we calculate the bounding box $B_f$ of the foreground and select a random patch. Then we resize the selected patch based on an area ratio (to the area of the background) which is sampled from a uniform distribution $U(\beta, 1)$. Finally, we ’paste’ the resized patch to a random location in the background.
2.2 Cross-view Saliency Alignment

Given two views $I_q$ and $I_k$ of Image $I$ augmented by a pipeline containing SaliencySwap and other augmentation operations such as random flipping and color jittering. Define $M_k$ and $M_q$ to be their saliency masks, respectively. Let $z^l_q$ and $z^l_k \in \mathbb{R}^{W \times H \times C^l}$ be the $C^l$ dimensional $H^l \times W^l$ feature maps encoded by an encoder network $z = f(I)$ (e.g., ResNet) truncated at stage $l$. We adopt two types of non-linear projector necks $g(\cdot)$ on top of the encoder to form a $d$-dim projection. The two-layer MLP projector generates the following projection $h_{mlp} = g_{mlp}(z)$ as proposed by SimCLR. Similarly, a convolutional projection $h_{conv}$ is given by a convolution projector $g_{conv}$, which consists of two $1 \times 1$ convolution layers with a batch normalization layer and a ReLU layer in between. Following BYOL, two predictor heads $p_{mlp}(\cdot)$ and $p_{conv}(\cdot)$, which have the same network structures except for different input dimensions, are adopted to match the output of one view to the other. Specially, $p_{conv}(\cdot)$ is designed for saliency alignment. Figure 2 (a) shows the overall framework.

**Cross-view Attention.** We seek to capitalize on the pixel-level foreground semantic interactions between the feature maps of two different augmented views. We first build a cross-view attention map:

$$A^l_{q,k} = p_{conv}(h^l_{q,conv}) \otimes h^l_{k,conv}^T,$$

where $A^l_{q,k}$ denotes the attention map is of view $k$ w.r.t. view $q$, $T$ denotes matrix transposition, and $\otimes$ denotes matrix multiplication. The location-aware attention map $A^l \in \mathbb{R}^{W^l \times H^l \times H^l}$ indicates a pair-wise spatial correspondence between any pixel from $p_{conv}(h^l_{q,conv})$ and any pixel from $h^l_{k,conv}$. Symetrically, we get the attention map $A^l_{k,q}$ of view $q$ w.r.t. view $k$ by interchanging $q$ and $k$ of Eqn. 3.

**Joint Saliency Alignment.** To enhance the encoder’s ability to identify the location of the foreground object, we propose to align the saliency mask with a correspondence intensity matrix that captures the pixel-level correlation from the feature map of one view to the other, as shown in Figure 2 (b). The correspondence intensity matrix $C$ is formulated as follows:

$$C^l_{q,k} = \text{max}(\sigma(A^l_{q,k})),$$

where $\sigma(\cdot)$ denotes sigmoid activation. Note that the shape of $C^l_{q,k}$ is $W^l \times H^l$ and the $\text{max}(\cdot)$ operation is performed over the second axis of $A^l_{q,k}$. We then define a symmetrized alignment loss between the saliency mask $M$ and the correspondence intensity matrix $C$:

$$\mathcal{L}_{\text{Align}} = ||\delta^l(M_q) - C^l_{q,k}||^2 + ||\delta^l(M_k) - C^l_{k,q}||^2,$$

where $\delta^l(\cdot) : \mathbb{R}^{W \times H} \rightarrow \mathbb{R}^{W^l \times H^l}$ denotes the bilinear downsampling operation. The proposed alignment loss restricts the most cross-view correlated pixels to the saliency region and thus gives the model localization ability. In addition, leveraging cross-layer semantics also enhances the representation of multi-scale learning.

**Joint Objective.** We define a contrastive loss $\mathcal{L}_{\text{Cont}}$ with the prediction vector $p_{q,mlp} \overset{\text{def}}{=} p_{mlp}(h_{q,mlp})$ and the projection vector $h_{k,mlp}$ using negative cosine similarity $D(\cdot)$ as:

$$\mathcal{L}_{\text{Cont}} = \frac{1}{2}D(p_{q,mlp}, h_{k,mlp}) + \frac{1}{2}D(p_{k,mlp}, h_{q,mlp}),$$

where $D(p, h) = -\frac{p}{||p||_2} \cdot \frac{h}{||h||_2}$. The joint objective for the second-stage pretext task is:

$$\mathcal{L}_{\text{CVSA}} = \mathcal{L}_{\text{Cont}} + \mathcal{L}_{\text{Align}}.$$
3 Experiments

3.1 Experimental settings

Implementation details. We use ResNet as the encoder $f$ followed by two projectors and two predictor sub-networks. During the first-stage pre-training, we follow the exact experimental setup in BYOL. For the second-stage in dual-stage pre-training, the model is initialized with the pre-trained weight from the first-stage and the first two stages of the ResNet backbone are frozen. We use a learning rate of $lr \times \text{BatchSize}/256$ with $\text{BatchSize} = 1024$ and a base $lr$ selected from $\{0.3, 0.6, 0.9, 1.2\}$. The embedding dimension is set to $d = 256$ as BYOL. As for image augmentations, we follow the settings in MoCo.v2 for all contrastive learning methods. We replace RandomResizedCrop (RRC) with the proposed SaliencySwap (SS) and adopt all other augmentations in MoCo.v2 [5], detailed in Appendix A. We grid search cropping scale ratio of SaliencySwap $\lambda \in \{0.08, 0.2, 0.5, 0.8\}$ and set $\lambda = 0.5$ by default. To balance the performance and computational cost, we adopt the stage $l = 4$ for the alignment. For simplicity, we use the ground truth bounding box of fine-grained datasets because most saliency detectors are trained in a supervised manner. An ablation study of the performance using different detection algorithms is given in Appendix B. We use the same pre-training setup as in Appendix A.1 for other unstated setups.

Dataset. We assess the performance of the representation pre-trained using dual-stage, first-stage only and second-stage only on four small-scale and one large-scale fine-grained benchmarks: 1) CUB-200-2011 [42] (CUB) contains 11,788 images from 200 wild bird species, 2) Stanford-Cars [21] (Cars) contains 16,185 images of 196 car subcategories, 3) FGVC-Aircraft [29] (Aircrafts) contains 10,000 images of 100 classes of aircrafts, 4) NA-birds [41] (NAbirds) is a large dataset with 48,562 images for over 555 bird classes and 5) iNaturalist2018 (iNat2018) [19] is an unbalanced long tail dataset which contains 437,513 images of 8,142 taxa coming from 14 super-classes. We follow the standard dataset partition in the original works. For the first-stage pre-training, we adopt two popular datasets: 1) ImageNet-1k (IN-1k) contains 1.28 million of training images. 2) MS-COCO (COCO) contains 118k images with more complex scenes of many objects.

Hyperparameters for CVSA. We search for the optimal hyperparameters for our proposed CVSA on the validation set of fine-grained datasets using ResNet-18 as the encoder. We set the batch size to 1024 and follow other training settings of BYOL [15]. The base learning rate is set to 0.3 for the Cars dataset and 0.6 for the other three datasets. The cropping scale ratio of SaliencySwap is $\lambda = 0.5$ by default. As for the stage $l$ in CVSA, we analyze the performance and the computational cost of using $l \in \{3, 4\}$ since the first two stages are frozen in the dual-stage. Furthermore, we find that using $l = 4$ achieves a balance between the performance and the computation cost.

3.2 Comparison with State-of-the-art

We choose two hand-crafted methods (Rel-Loc and Rot-Pred), three commonly used contrastive learning methods (SimCLR, MoCo.v2, and BYOL), and three extension methods (LooC*, DiLo, and InsLoc) for comparison. DiLo and InsLoc are reproduced using official code, with other methods reproduced using OpenMixup [24]. LooC* denotes its rotation version reproduced by us. Since our approach extends BYOL, we choose BYOL as the baseline. Notice that grey indicates baseline, green denotes improvement over baseline while red for degradation performance, bold denotes the best performance.
DUAL-STAGE
We evaluate CVSA on large-regarded as the baseline by default.

In a word, CVSA and formulation in Appendix A.2 that discriminative feature extraction relies on the size of the performance gain is affected by the size of dataset used. This is consistent with our regarded as the baseline by default.

Small-scale scenarios. We perform 800 epochs pre-training with ResNet-50 encoder using different pre-training stage settings on four small-scale fine-grained datasets and report the top-1 classification accuracy under the fine-tune protocol. As shown in Table 1, when performing the second-stage only setting, namely pre-training with the training set of fine-grained datasets from scratch, our proposed CVSA outperforms BYOL by a large margin on all benchmarks. When applying the dual-stage setting using COCO for the first-stage, as shown in Table 2, CVSA shows a consistent improvement on representations learned during the first-stage while most comparing methods yield worse performance than the first-stage pre-trained BYOL baseline. However, when using ImageNet for the first-stage pre-training, as shown in Table 3, the improvement of CVSA in the dual-stage setting is not as significant on Aircraft and Cars. We hypothesize that this is due to the iconic nature of these two datasets. We find that most images of these two datasets are of similar and straightforward backgrounds, restricting the background variation imposed by our approach. Besides, comparison of Table 1 with Table 2 and 3 suggests that the performance gain is affected by the size of dataset used. This is consistent with our formulation in Appendix A.2 that discriminative feature extraction relies on the size of the pre-training dataset, and localization ability alone is not enough for fine-grained recognition. In a word, CVSA and dual-stage pre-training pipeline improve learned representation for the small-scale fine-grained classification.

Large-scale scenarios. Next, we evaluate CVSA on large-scale benchmark iNat2018 with the dual-stage pre-training. Following MoCo [17], we adopt the linear evaluation protocol as mentioned in Appendix A.1 training

<table>
<thead>
<tr>
<th>Methods</th>
<th>CUB</th>
<th>NAbirds</th>
<th>Aircrafts</th>
<th>Cars</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>58.51</td>
<td>(6.34)</td>
<td>65.78</td>
<td>(6.76)</td>
</tr>
<tr>
<td>ReL-Loc</td>
<td>65.89</td>
<td>(1.04)</td>
<td>72.60</td>
<td>(0.06)</td>
</tr>
<tr>
<td>Rot-Pred</td>
<td>66.67</td>
<td>(1.82)</td>
<td>73.01</td>
<td>(0.47)</td>
</tr>
<tr>
<td>SimCLR</td>
<td>63.43</td>
<td>(1.42)</td>
<td>72.05</td>
<td>(0.49)</td>
</tr>
<tr>
<td>MoCo.v2</td>
<td>63.21</td>
<td>(1.64)</td>
<td>71.36</td>
<td>(1.18)</td>
</tr>
<tr>
<td>LooC*</td>
<td>66.42</td>
<td>(1.57)</td>
<td>72.84</td>
<td>(0.30)</td>
</tr>
<tr>
<td>InsLoc</td>
<td>64.87</td>
<td>(0.02)</td>
<td>72.80</td>
<td>(0.26)</td>
</tr>
<tr>
<td>BYOL</td>
<td>64.85</td>
<td>(0.00)</td>
<td>72.54</td>
<td>(0.00)</td>
</tr>
<tr>
<td>BYOL+DiLo</td>
<td>66.16</td>
<td>(1.131)</td>
<td>73.12</td>
<td>(0.58)</td>
</tr>
<tr>
<td>BYOL+CVSA</td>
<td>66.88</td>
<td>(2.03)</td>
<td>73.75</td>
<td>(1.21)</td>
</tr>
</tbody>
</table>

Table 1: Comparison of dual-stage only setting on fine-grained benchmarks. Top-1 accuracy (%) under fine-tuned evaluation is reported. Random denotes random ini- is reported. The first-stage is pre-trained on tialized models in the second-stage. BYOL is COCO, and ✓ denotes performing the second-stage pre-trained on fine-grained datasets.

<table>
<thead>
<tr>
<th>Methods</th>
<th>CUB</th>
<th>NAbirds</th>
<th>Aircrafts</th>
<th>Cars</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ref-Loc</td>
<td>✓</td>
<td>67.33</td>
<td>(1.22)</td>
<td>73.82</td>
</tr>
<tr>
<td>Rot-Pred</td>
<td>✓</td>
<td>67.75</td>
<td>(0.80)</td>
<td>74.26</td>
</tr>
<tr>
<td>SimCLR</td>
<td>✓</td>
<td>68.30</td>
<td>(0.38)</td>
<td>73.51</td>
</tr>
<tr>
<td>MoCo.v2</td>
<td>✓</td>
<td>68.47</td>
<td>(0.08)</td>
<td>73.73</td>
</tr>
<tr>
<td>BYOL</td>
<td>✓</td>
<td>67.60</td>
<td>(1.05)</td>
<td>73.18</td>
</tr>
<tr>
<td>BYOL+DiLo</td>
<td>✓</td>
<td>68.71</td>
<td>(0.16)</td>
<td>74.45</td>
</tr>
<tr>
<td>BYOL+CVSA</td>
<td>✓</td>
<td>68.91</td>
<td>(0.15)</td>
<td>74.67</td>
</tr>
</tbody>
</table>

Table 2: Comparison of dual-stage pre-training on fine-grained benchmarks. Top-1 accuracy (%) under fine-tuned evaluation is reported. The first-stage is pre-trained on IN-1k, and ✓ denotes performing the second-stage pre-training on corresponding fine-grained datasets. Supervised denotes the supervised pre-training on IN-1k.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Stage 2</th>
<th>CUB</th>
<th>NAbirds</th>
<th>Aircrafts</th>
<th>Cars</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised</td>
<td>✓</td>
<td>81.02</td>
<td>80.09</td>
<td>87.25</td>
<td>90.61</td>
</tr>
<tr>
<td>SimCLR</td>
<td>✓</td>
<td>73.99</td>
<td>(2.64)</td>
<td>76.30</td>
<td>(2.59)</td>
</tr>
<tr>
<td>MoCo.v2</td>
<td>✓</td>
<td>73.19</td>
<td>(3.44)</td>
<td>75.64</td>
<td>(3.25)</td>
</tr>
<tr>
<td>MoCo.v2</td>
<td>✓</td>
<td>71.77</td>
<td>(4.86)</td>
<td>73.96</td>
<td>(4.93)</td>
</tr>
<tr>
<td>InsLoc</td>
<td>✓</td>
<td>75.83</td>
<td>(0.80)</td>
<td>78.86</td>
<td>(0.83)</td>
</tr>
<tr>
<td>BYOL</td>
<td>✓</td>
<td>76.63</td>
<td>(0.00)</td>
<td>78.89</td>
<td>(0.00)</td>
</tr>
<tr>
<td>BYOL</td>
<td>✓</td>
<td>72.46</td>
<td>(4.17)</td>
<td>76.12</td>
<td>(2.77)</td>
</tr>
<tr>
<td>BYOL+DiLo</td>
<td>✓</td>
<td>76.60</td>
<td>(0.03)</td>
<td>79.04</td>
<td>(0.15)</td>
</tr>
<tr>
<td>BYOL+CVSA</td>
<td>✓</td>
<td>77.10</td>
<td>(0.47)</td>
<td>79.64</td>
<td>(0.75)</td>
</tr>
</tbody>
</table>

Table 3: Comparison of dual-stage pre-training on fine-grained benchmarks. Top-1 accuracy (%) under fine-tuned evaluation is reported. The first-stage is pre-trained on IN-1k, and ✓ denotes performing the second-stage pre-training on corresponding fine-grained datasets. Supervised denotes the supervised pre-training on IN-1k.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Stage 1</th>
<th>Stage 2</th>
<th>Sup.</th>
<th>MoCo.v2</th>
<th>BYOL</th>
<th>BYOL+DiLo</th>
<th>BYOL+CVSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>BYOL</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BYOL+DiLo</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BYOL+CVSA</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Comparison of dual-stage pre-training on iNat2018. Top-1 accuracy (%) under linear evaluation is reported. Sup. denotes the supervised pre-training on iNat2018 in the second-stage.
we study the effect when freezing various stages of ResNet-18 in the texture is no longer the only clue for contrastive pre-training to learn representation. Then, pre-trained on COCO in the first-stage initialize the model of all methods in the second-stage.

Figure A1 left, the performance of BYOL fluctuates drastically under the different scaling factors of SS during training. The first-stage is pre-trained second-stage. Right: the effect of freezing different ResNet stages during second-stage. The dotted grey line indicates the performance of BYOL pre-trained on COCO during first-stage.

100 epochs with the basic learning rate $lr = 0.025$ and batch size of 256. In Table 4, CVSA outperforms existing methods in both settings indicating its effectiveness in the large-scale datasets, especially improving the baseline by 0.9% with second-stage-only. Compared with the results in Table 3, the performance gain is more significant than second-stage pre-training on small-scale datasets like Aircraft and Cars, which suggests that CVSA benefits from a larger data size. Meanwhile, using dual-stage pre-training with IN-1k for the first-stage helps contrastive methods to learn better representations than the second-stage only setting.

### 3.3 Ablation Study

We perform 400 epochs pre-training with ResNet-18 on CUB for the first three ablation studies. As for the fourth ablation for the dual-stage pre-training, we perform 200 epoch first-stage pre-training on large datasets and 400 epochs second-stage pre-training on target datasets. The top-1 accuracy under the fine-tune evaluation is reported. Appendix B provides more results.

**Module effectiveness ablation.** We demonstrate the effectiveness of our CVSA by adding modules one by one onto the baseline. We compare the performance of MoCo.v2 and BYOL using SS against RandomResizedCrop (RRC) while keeping all other augmentations unchanged. As shown in Figure 3 left, SS outperforms RRC (using the default scaling factor of 0.08) both for MoCo.v2 and BYOL. We observe a further performance improvement from adding the saliency alignment loss (Align) onto BYOL using SS. Now, we have shown that both SS and Align contribute to higher performance.

**Hyperparameter ablation.** We then analyze the performance of SS using different crop scaling factors, specifically $\lambda \in \{0.08,0.2,0.3,0.4,0.5,0.8\}$ in Figure 3 left. From Figure A1 left, the performance of BYOL fluctuates drastically under the different scaling factors of RRC, with the best result achieved with a scaling factor being 0.08. However, BYOL yields similar performance under different choices of $\lambda$ of SS. We argue that SS, together with Align, helps the model to localize on the foreground object, and thus local texture is no longer the only clue for contrastive pre-training to learn representation. Then, we study the effect when freezing various stages of ResNet-18 in the second-stage. We initialize the model of all methods in the second-stage with the weights of BYOL baseline pre-trained on COCO in the first-stage. In Figure 3 right, the horizontal axis indicates freezing up to different stages of ResNet-18. The best performance is reached when freezing up...
to the second-stage of ResNet. The early stages of ResNet mostly extract low-level texture information, while the later stages are for higher-level features such as discrimination and localization. Intuitively, we wish to enhance the localization ability without jeopardizing the texture extraction ability acquired during the first-stage. This explains the reason for freezing the first two stages during the second-stage, which is common in object detection [34].

**Background dataset ablation.** Moreover, we compare different foreground and background fusion methods (SS and DiLo) based on CUB dataset using second-stage only settings. As shown in Table 3.3, replacing the background with other datasets usually results in performance degradation, while fusing with complex backgrounds might improve the performance. We hypothesize that exploring task-specific backgrounds is more critical in fine-grained scenarios. Meanwhile, it indicates that the proposed CVSA (SS+Align) can well explore the background of CUB.

**First-stage pre-training ablation.** Additionally, we verify the necessity of the first-stage learning discrimination on iconic datasets in the dual-stage pre-training scheme. We compare naive BYOL and CVSA for the first-stage pre-training on IN-1k, iNat2018, and COCO. As shown in Table 5, using IN-1k (iconic) for the first-stage yields better performance than scenic datasets, indicating the necessity of the first-stage on iconic datasets. Since CVSA is design for the second-stage on target datasets, we find that using CVSA on second-stage outperforms naive BYOL while producing degraded performance on the first-stage.

**Visualization of localization abilities.** Lastly, we compare the localization abilities by Grad-CAM [35] visualization of SS (BYOL+CVSA) and RRC (BYOL). Figure 4 (a) and (b) shows that SS enables the model to better localize the fine-grained target than RRC. Comparing to Figure 4 (a)(b), Figure 4 (c) shows that dual-stage training further improves localization than the first-or second-stage only.

<table>
<thead>
<tr>
<th>Background</th>
<th>BYOL+DiLo</th>
<th>BYOL+SS</th>
<th>BYOL+SS+Align</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUB</td>
<td>64.14 (+0.00)</td>
<td>64.35 (+0.00)</td>
<td>65.02 (+0.00)</td>
</tr>
<tr>
<td>COCO</td>
<td>60.07 (+0.07)</td>
<td>60.42 (+0.93)</td>
<td>61.23 (-3.79)</td>
</tr>
<tr>
<td>NAbirds</td>
<td>62.51 (+1.15)</td>
<td>63.06 (+1.29)</td>
<td>64.80 (+0.22)</td>
</tr>
<tr>
<td>CUB+COCO</td>
<td>61.26 (+0.88)</td>
<td>61.49 (+2.86)</td>
<td>62.01 (+3.01)</td>
</tr>
<tr>
<td>CUB+NAbirds</td>
<td>64.28 (+0.14)</td>
<td>64.23 (-0.12)</td>
<td>65.50 (+0.48)</td>
</tr>
</tbody>
</table>

Table 6: Evaluation of background datasets extension for the second-stage pre-training.

We study the effect of fusing different background datasets on CUB.

Figure 4: Grad-CAM visualization of BYOL and BYOL+CVSA on CUB200.

4 Related Work

Self-supervised methods have largely reduced the performance gap between supervised models on various downstream vision tasks. Most early methods design hand-crafted pretext tasks [10, 13, 56]. For example, Gidaris et al. [13] proposed learning image features by training the network to predict different rotation angles of the same image. Doersch et al. [10] proposed learning image features by training the network to predict the correct order of random cut image patches. These pretext tasks rely on somewhat ad-hoc heuristics, which limits the generality of learned representations.
Recently, contrastive learning [3, 4, 5, 17, 23, 51, 52] achieved state-of-the-art performance, which learns instance-level discriminative representations by contrasting positive pairs against negative pairs. In particular, MoCo [17] adopts a memory bank to store negative samples while embedding different views of the same image using an online encoder and a momentum encoder. SimCLR [4] yields comparable results using the sufficiently large batch size in replacement of the memory bank. Getting rid of the notion of negative pairs, BYOL [15] pulls together positive pairs generated by online and momentum encoders with the help of a predictor and the stop gradient mechanism. Another popular form of self-supervised learning is clustering-based methods [2, 3, 54]. Without computing pairwise comparisons, SwAV [3] maps image features to a set of learnable prototype vectors. Various mechanisms are proposed to learn useful representations rather than trivial solutions in contrastive methods such as a constant representation.

More recently, some research endeavors have been made on top of contrastive methods to enhance pre-training quality for specific downstream tasks, such as object detection and segmentation [25, 26, 45, 47, 48]. Most methods adopts detection or segmentation components to learn pixel-level contrastive representation [8, 11, 37, 49]. DSC [25] proposed to model pixel-level semantic structures within images by taking into consideration the semantic relations of both intra- and inter-image pixels. Self-EMD [26] learns representations by measuring the similarity among all location pairs using the earth mover’s distance (EMD). LooC [39] proposed to construct separate embedding sub-spaces for each augmentation instead of a single embedding space. DiLo [57] proposed a copy-paste [12, 33] based augmentation approach that randomly pastes masked foreground onto a variety of backgrounds. CASTing [37] crops views based on a ratio threshold of the area of saliency regions (required mask-level supervision) to the area of the cropped patch. However, existing methods are trained on general-purpose coarse-grained datasets while neglecting fine-grained scenarios where low-level background texture features provide little clue to the category information of the foreground subject. To address this issue, we propose a dual-stage pre-training pipeline that utilizes coarse- and fine-grained datasets for better fine-grained representation learning. Appendix C provides a discussion of the relationship between our proposed CVSA and previous methods.

Current efforts [9, 20, 38, 46, 58, 59] in fine-grained recognition are primarily dedicated to fine-tuning model pre-trained on supervised ImageNet either by localizing distinct parts or by learning fine-grained features. However, there exists little exploration in self-supervised pre-training for fine-grained categorization. In the paper, we attempt to bring localization and fine-grained feature representation learning to the pre-training stage, using fine-grained datasets. To address this issue, we propose a dual-stage pre-training pipeline that utilizes coarse- and fine-grained datasets for better fine-grained representation learning.

5 Conclusion

In this paper, we find that learning to extract discriminative texture information and localization are equally crucial for fine-grained self-supervised pre-training with empirical analysis. We proposed a dual-stage pre-training pipeline with the first-stage to train feature extraction and the second-stage to train localization. To empower the model with localization abilities in the second-stage, we propose cross-view saliency alignment (CVSA), a new unsupervised contrastive learning framework. Extensive experiments on fine-grained benchmarks demonstrate the effectiveness of our contributions in learning better fine-grained representations.
Acknowledgement

This work is supported by the Science and Technology Innovation 2030- Major Project (No. 2021ZD0150100) and the National Natural Science Foundation of China (No. U21A20427). We thank Zicheng Liu, Yifan Zhao, and all reviewers for polishing the writing.

References


[58] Yifan Zhao, Ke Yan, Feiyue Huang, and Li Jia. Graph-based high-order relation discovery for fine-grained recognition. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021.
