

Prototype-based Contrastive Learning with Stage-wise Progressive Augmentation for Self-Supervised Fine-Grained Learning

Baofeng Tan¹ Xiu-Shen Wei^{2*} Lin Zhao¹

¹School of Computer Science and Engineering, Nanjing University of Science and Technology

²School of Computer Science and Engineering, and Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications, Southeast University

Abstract

In this paper, we mitigate the problem of Self-Supervised Learning (SSL) for fine-grained representation learning, aimed at distinguishing subtle differences within highly similar subordinate categories. Our preliminary analysis shows that SSL, especially the multi-stage alignment strategy, performs well on generic categories but struggles with fine-grained distinctions. To overcome this limitation, we propose a prototype-based contrastive learning module with stage-wise progressive augmentation. Unlike previous methods, our stage-wise progressive augmentation adapts data augmentation across stages to better suit SSL on fine-grained datasets. The prototype-based contrastive learning module captures both holistic and partial patterns, extracting global and local image representations to enhance feature discriminability. Experiments on popular fine-grained benchmarks for classification and retrieval tasks demonstrate the effectiveness of our method, and extensive ablation studies confirm the superiority of our proposals. Codes are available at <https://github.com/SEU-VIPGroup/PAPN>.

1. Introduction

Self-Supervised Learning (SSL) has become a cornerstone in computer vision, empowering models to extract meaningful features from vast amounts of unlabeled data [5, 14]. By leveraging pretext tasks [48] that capitalize on the inherent structure of the data, SSL facilitates the pre-training of deep neural networks, mitigating the reliance on annotated datasets. SSL has catalyzed substantial progress in various computer vision applications, such as image classification [2, 13, 55], image retrieval [34, 36, 44], object detection [23, 32, 51], and semantic segmentation [26, 42, 56].

However, despite its effectiveness in generic category recognition, SSL faces significant challenges when applied to fine-grained visual representation learning (FGVR) [31, 45, 46, 50, 57]. FGVR requires the identification of minute details and subtle nuances, such as the slight differences in a bird’s beak or the texture of a car’s finish-details that are often overlooked by more generalized methods of SSL. Although recent advancements in SSL, such as the multi-stage alignment strategy [17, 54], which aims to align features generated at different stages of the query encoder and the key encoder, has shown promise in enhancing SSL, its effectiveness in fine-grained scenarios remains an open question.

Our preliminary experiments revealed that while multi-stage alignment excelled with generic datasets like *Caltech-256* [10], its performance markedly deteriorated with fine-grained datasets such as *CUB200* [40], exacerbating the disparity between self-supervised and supervised training outcomes. Further investigation indicated that the root cause lies in the data augmentation strategy employed in multi-stage alignment, which is not suitable for the fine-grained setting. Specifically, the semantic information within embeddings at shallow stages is prone to distortion due to strong data augmentation, leading to semantic misalignments and a subsequent decline in SSL performance.

Based on the aforementioned findings, we propose the Progressive Augmentation Prototype-based Network (PAPN) method, which consists of two core components: Stage-wise Progressive Augmentation (SPA) and Prototype-based Holistic-Partial Contrastive Learning (PHP). Since the data augmentation used by multi-stage alignment fails to maintain semantic consistency in fine-grained settings, the SPA module implements a progressive augmentation that uses different augmentation intensities for different stages to preserve semantic information, thereby enhancing SSL performance. Additionally, recognizing the critical importance of fine-grained object parts, such as a bird’s head or tail, which are often overlooked by existing SSL methods, the PHP module uses prototype clustering to capture and model these parts end-to-end. This enables holistic-partial SSL, significantly enhancing the discriminative power of the learned

*Corresponding author. This work was supported by National Natural Science Foundation of China under Grant (62522602, 62272231, 62172222), CIE-Tencent Robotics X Rhino-Bird Focused Research Program, and the Fundamental Research Funds for the Central Universities (4009002401). This research work is supported by the Big Data Computing Center of Southeast University.

representations. Collectively, these two modules alleviate the problem of semantic misalignment and improve the effectiveness of SSL on FGVR tasks.

To evaluate our method, we conduct extensive experiments using popular fine-grained benchmarks, including *CUB200* [40], *Cars* [20], *Aircraft* [27] and *iNat2019* [1]. Quantitative results of classification and retrieval accuracies on these datasets show that the proposed PAPN method consistently and significantly outperforms existing state-of-the-art methods. Ablation studies of the crucial modules in PAPN also validate their individual effectiveness. Furthermore, qualitative visualization results confirm that our method successfully extracts distinct part information from images, reinforcing the effectiveness of our method.

In summary, our work has a threefold contribution:

- We identify the failure of SSL on fine-grained datasets, mitigating the challenge of self-supervised fine-grained visual representation learning.
- We propose the Progressive Augmentation Prototype-based Network method, comprising two pivotal modules: Stage-wise Progressive Augmentation and Prototype-based Holistic-Partial Contrastive Learning.
- We conduct experiments on multiple fine-grained datasets, validating the effectiveness of the proposed method from both quantitative and qualitative perspectives.

2. Related Work

2.1. Self-Supervised Contrastive Learning

Self-supervised learning (SSL) [12] is a popular technique for learning representations from unlabeled data, with contrastive learning (CL) becoming a dominant method [4, 11, 28]. The process of CL can be abstracted as a dictionary query task [14], where the core idea is to attract the query to its positive keys while repelling it from negative samples.

During training, a sample is selected from the dataset and embedded as \mathbf{q} by an encoder f_q . An augmented version of this sample is embedded as \mathbf{k}_+ by another encoder f_k , serving as the positive view. All other key embeddings within the same batch are considered negative samples. The goal is to train the model to learn a representation that effectively discriminates between positive and negative samples.

Based on this idea, recently, several methods such as SimCLR [5], MoCo [14], and MoCo v2 [7] have been proposed, achieving results on *ImageNet* [9] that are comparable to state-of-the-art supervised methods. SimCLR [5] utilized negative samples that coexist within the current batch, which necessitated a large batch size for effective performance. In contrast, MoCo [14] introduced a queue-based mechanism to maintain a queue of negative samples, and it employed a momentum encoder for one of the branches, thereby enhancing the consistency of negative samples. Building upon MoCo, MoCo v2 [7] demonstrated that the quality of learned

representations can be significantly improved by selecting robust pretext tasks and employing suitable augmentations.

Although existing contrastive learning methods have demonstrated reasonable performance on generic datasets, they still face challenges in learning discriminative representations in fine-grained settings, which can be attributed to the “coarse-grained bias” mentioned in previous works [8].

2.2. Fine-Grained Visual Representation Learning

Fine-grained visual representation learning (FGVR) [45] focuses on learning discriminative features to distinguish between visually similar subordinate categories. Traditional supervised approaches achieve this by explicitly localizing discriminative parts [39, 49], encoding higher-order feature statistics via models like Bilinear CNNs [24, 25], or incorporating external information [15].

However, these supervised methods are constrained by their reliance on costly fine-grained labels. To mitigate this, recent self-supervised FGVR methods have been developed [38, 43, 47]. A common strategy is to leverage “prior knowledge” for guidance. For instance, LCR [38] and LDF [43] utilize attention maps from Grad-CAM [33] as pseudo-supervision to direct the model’s focus toward relevant regions, while CVSA [47] aligns features with pre-trained saliency detectors.

In contrast to these methods that require saliency or CAM maps as “prior knowledge”, our method does not rely on any such external guidance. Instead, it focuses on redesigning the data augmentation strategy and employs holistic-partial contrastive learning to directly enhance the self-supervised fine-grained visual representation capability.

3. Preliminary Results

Observations Presently, the majority of CL methods [5, 6, 11, 14] concentrate on attracting positive views while repelling negative samples on a hypersphere, thereby increasing the angular separation between different samples to improve model performance [41]. Additionally, several recent studies [17, 54] have adopted the strategy of multi-stage alignment. This strategy focuses on aligning features across various stages of the encoders f_q and f_k , thereby enhancing the representational power of CL. To empirically evaluate the impact of multi-stage alignment on CL, we conducted an experiment comparing supervised training with the widely used contrastive learning method MoCo v2 [7] on the generic dataset *Caltech-256* [10].

To assess the separation of representations in CL, we calculate the average angle between the normalized embeddings of each sample and all other samples throughout the training process. The results are depicted in Figure 1a. On the generic dataset, both supervised training and MoCo v2 showed a gradual increase in the average angle over the course of training, with a negligible final difference between

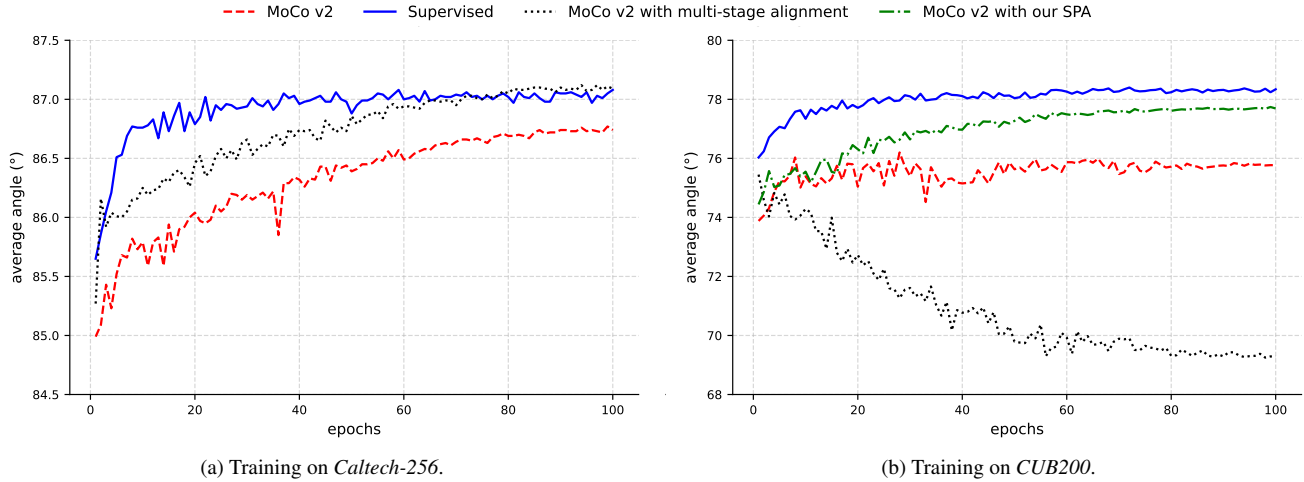


Figure 1. The average angle between embeddings of different samples on the generic dataset *Caltech-256* and the fine-grained dataset *CUB200*. The vertical axis ranges from 0 to 180 degrees. Multi-stage alignment proves beneficial for enhancing MoCo v2 on the generic dataset but encounters challenges on the fine-grained dataset. Our proposed Stage-wise Progressive Augmentation (SPA) module effectively alleviates this issue, as detailed in Section 4.

them. This suggests that, in CL, the embeddings of different samples are repelled on a hypersphere, underscoring the efficacy of CL. Notably, when multi-stage alignment was incorporated into MoCo v2, the difference between supervised and contrastive learning was further reduced and even surpassed the supervised level during the final epochs, indicating the effectiveness of multi-stage alignment on the generic dataset. These findings raised a question: Is using multi-stage alignment equally effective in enhancing MoCo v2 on fine-grained datasets?

As shown in Figure 1b, when the same methods were applied to fine-grained datasets such as *CUB200* [40], the results diverged significantly. For MoCo v2, the average angle curve was markedly lower compared to that of supervised training, with neither the value nor the growth rate being comparable. This phenomenon aligns with the concept of “coarse-grained bias”, an issue well-documented in previous works [8, 16]. However, interestingly, unlike observations on the generic dataset, multi-stage alignment did not enhance the performance of MoCo v2 but instead resulted in a decline. That is, when dealing with fine-grained images, multi-stage alignment not only failed to increase the angle of different samples on a hypersphere but instead encouraged them to cluster closely. This overarching observation begs the pivotal question: *What are the root causes that lead to the failure of multi-stage alignment on fine-grained data?*

Conjecture & Discussions In fine-grained settings, small sample differences result in small angles between embeddings on a hypersphere, making them prone to semantic misalignment under strong data augmentation. Recent studies, like [16], also emphasize the importance of data augmentation for CL, especially in fine-grained scenarios. Thus, we

Algorithm 1 Linear Augmentation Pseudocode

Input: Intensity s of data augmentation; Input image img .

Output: Augmented image aug_img .

- 1: **BEGIN**
 - 2: **Initialize augmentation sequence:**
 - 3: $\# s \in [0, 1]$, representing the intensity of augmentation.
 - 4: RandomResizedCrop(size=224, scale=(1.0 - 0.8s, 1.0))
 - 5: RandomGrayscale(probability=0.2s)
 - 6: ColorJitter(params=(0.2s, 0.2s, 0.2s, 0.1s), probability=0.8s)
 - 7: GaussianBlur(sigma=(0.1s, 2.0s), probability=0.5s)
 - 8: RandomHorizontalFlip()
 - 9: ToTensor()
 - 10: Normalize(mean, std)
 - 11: $aug_img \leftarrow$ Apply augmentation sequence to img
 - 12: **RETURN** aug_img
 - 13: **END**
-

conjecture: *Does the data augmentation strategy used by multi-stage alignment alter semantic information at different stages, thereby impeding CL?*

To substantiate this hypothesis, we conducted an experiment on the *CUB200* dataset, involving the application of a set of increasing data augmentation intensities at different stages of the backbone to explore semantic changes.

Firstly, we define q_l^s as the pooled query vector from the l stage of the encoder f_q , and k_l^s as the positive view from f_k . The variable s represents the intensity of data augmentation applied to the input image, where $s \in [0, 1]$. At $s = 0$, the augmentation includes only horizontal flipping. As s increases within $(0, 1)$, we linearly enhance augmentation hyperparameters, such as the probability of ColorJitter, to achieve intermediate augmentation levels. At $s = 1$, the augmentation is equivalent to the original MoCo v2, representing the strongest augmentation level. The details are shown in Algorithm 1.

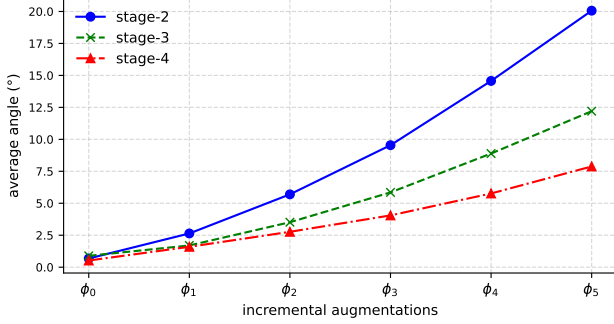


Figure 2. The average angle between embeddings and their positive views, generated by different stages of f_q and f_k . The horizontal axis represents a set of incremental augmentations defined in Section 3. As the intensity of augmentations increases, the average angles at different stages steadily rise.

Subsequently, we utilize ResNet-50 [13] as the encoder for both f_q and f_k , focusing on the last three stages, *i.e.*, stages 2, 3, and 4. We denote a set of incremental augmentations as $\{\phi_0, \phi_1, \phi_2, \phi_3, \phi_4, \phi_5\}$, where $\phi_i = \{\{q_2^{0.2i}, k_2^{0.2i}\}, \{q_3^{0.2i}, k_3^{0.2i}\}, \{q_4^{0.2i}, k_4^{0.2i}\}\}$. For each ϕ_i , we randomly select 5,000 training images to compute the average angle between $q_j^{0.2i}$ and $k_j^{0.2i}$ for $j \in \{2, 3, 4\}$. Note that, unlike Figure 1, we calculate the angle between embeddings of the sample and their positive views, rather than the sample and other samples. The results are illustrated in Figure 2. It can be clearly observed that, as data augmentation was continuously enhanced, the average angle across all stages increased steadily. This indicates that there are varying degrees of semantic shift between q_i^s and k_i^s produced by each stage. Moreover, it was worth noting that for deep stages, the change amplitude was small, while for shallow stages, the change amplitude was drastic, which reflected the characteristics of CNNs: Deeper stages exhibit the stronger capacity for abstracting semantic information, making them more tolerant to augmentation than shallower stages [53].

Under the fine-grained setting, multi-stage alignment aligns q_i^s and k_i^s across different stages to enhance CL. However, the experimental result indicates that when the positive view is augmented using MoCo v2’s data augmentation, denoted as ϕ_5 , the semantic offset between q_i^s and k_i^s at the shallow stage becomes significant, leading to a semantic misalignment. This confirms our conjecture. Moreover, from the perspective of Expectation Maximization (EM) optimization [22], this misalignment hinders k_+ from acting as the positive view to q . Consequently, the model attempts to pull q towards the negative samples k_- , which misguides the optimization process and consequently degrades the model’s performance. We provide more mathematical details in the supplementary material.

To ensure semantic consistency during multi-stage alignment, we propose a novel Stage-wise Progressive Augmentation (SPA) module. This module employs varying intensities

of data augmentation at different stages to obtain q_i^s and k_i^s for semantic alignment. The detailed implementation is described in Section 4.1. Upon applying this method, as illustrated by “MoCo v2 with our SPA” in Figure 1b, the average angle exhibited a steady increase, thereby alleviating the problem and validating our module’s effectiveness.

4. Methodology

Building on the conjecture presented in Section 3, we introduce the Progressive Augmentation Prototype-based Network (PAPN), which comprises two pivotal modules: Stage-wise Progressive Augmentation (SPA) and Prototype-based Holistic-Partial Contrastive Learning (PHP). SPA is designed to ensure the semantic consistency of each stage in encoders f_q and f_k during semantic alignment. Furthermore, acknowledging the critical role of local parts in fine-grained images [16], PHP employs prototype clustering to extract part information and implements holistic-partial contrastive learning to enhance the efficacy of contrastive learning. Detailed descriptions are provided in Figure 3.

4.1. Stage-wise Progressive Augmentation

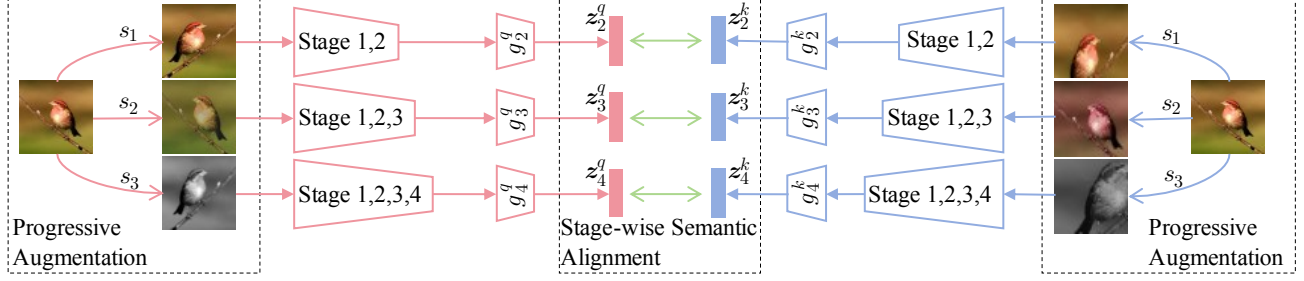
Preliminary results in Section 3 indicate that the data augmentation strategy employed by multi-stage alignment can alter semantic information at various encoder stages. To mitigate the risk of semantic deviation caused by inappropriate augmentation, we introduce a novel progressive augmentation strategy in this module. This strategy ensures semantic consistency at different stages during the training process. Subsequently, we apply stage-wise semantic alignment across stages, thereby enhancing the performance of contrastive learning.

Progressive Augmentation As discussed in Section 3, different training stages exhibit varying tolerance to data augmentation. To prevent semantic shifts that could occur from applying inappropriate augmentation, we implement progressive augmentation during the training process.

Formally, let us assume that the backbone f consists of L stages. Here, we consider the outputs of the last T stages. We follow the symbols q_l^s and k_l^s defined in Section 3, where $l \in \{L - T + 1, L - T + 2, \dots, L\}$, s denotes the intensity of data augmentation, and $q_l^s \in \mathbb{R}^{C_l}$, where C_l is the number of output channels at the l stage. To accommodate varying augmentation tolerances at different stages shown in Figure 2, we define a set of progressively increasing values $\{s_1, s_2, \dots, s_T\}$ to represent the intensity of data augmentation as each stage deepens.

During training, we first start with the smallest value s_1 in $\{s_1, s_2, \dots, s_T\}$ to obtain the pair $\{q_{L-T+1}^{s_1}, k_{L-T+1}^{s_1}\}$ at the $L - T + 1$ stage, and then progressively use the increased values in $\{s_1, s_2, \dots, s_T\}$

Stage-wise Progressive Augmentation



Prototype-based Holistic-Partial Contrastive Learning

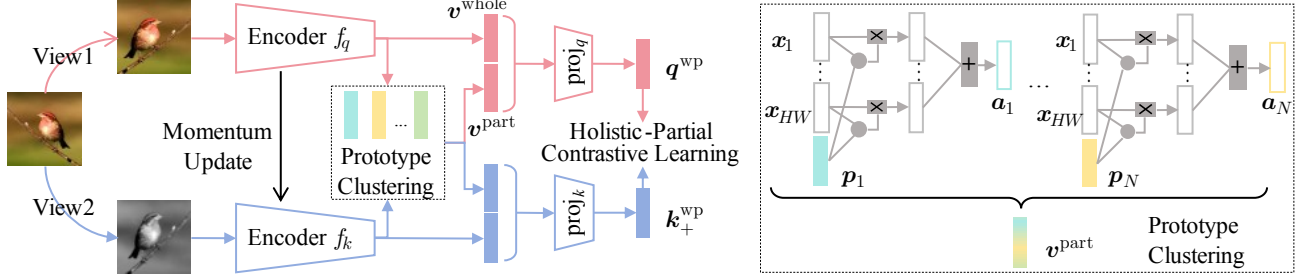


Figure 3. Overview of the proposed Progressive Augmentation Prototype-based Network (PAPN) method, which integrates two pivotal modules: Stage-wise Progressive Augmentation (SPA) and Prototype-based Holistic-Partial Contrastive Learning (PHP). The pink branch depicts the query encoder f_q , while the blue branch represents the key encoder f_k . In the Prototype Clustering, x_i denotes the descriptor from the last stage of the encoder, p_i represents the learnable prototype, a_i signifies the corresponding part feature, and the gray circle indicates the dot product following ℓ_2 normalization.

to obtain pairs produced by deeper stages (such as $\{q_L^{sT}, k_L^{sT}\}$). This process yields a set of pairs $\beta = \{\{q_{L-T+1}^{s1}, k_{L-T+1}^{s1}\}, \{q_{L-T+2}^{s2}, k_{L-T+2}^{s2}\}, \dots, \{q_L^{sT}, k_L^{sT}\}\}$. The goal is to align $\{q_{L-T+i}^{s_i}, k_{L-T+i}^{s_i}\}$ for i range from 1 to T semantically, thereby enhancing the consistency between the features extracted by f_k and f_q .

Compared to the original strategy, which uses the same intensity for augmentation across all stages, our progressive augmentation strategy enables the model to mitigate the semantic inconsistency problem, particularly when features are semantically aligned, especially in fine-grained scenarios.

Stage-wise Semantic Alignment The purpose of CL is to attract embeddings and their positive views while repelling embeddings of different samples on a hypersphere. To narrow the distance between embeddings and their positive views, we perform a comprehensive alignment of the output at each stage, thereby significantly enhancing the performance of contrastive learning.

Concretely, we aim to impose an alignment loss between pairs β obtained by the progressive augmentation strategy. To this end, we introduce a simple fully connected layer g_{L-T+i}^q at the $L - T + i$ stage that takes the $q_{L-T+i}^{s_i}$ as input and reduces it to a vector z_{L-T+i}^q of dimension D , similar to the layer g_{L-T+i}^k and the vector z_{L-T+i}^k . Following that, the set β becomes

$\{\{z_{L-T+1}^q, z_{L-T+1}^k\}, \{z_{L-T+2}^q, z_{L-T+2}^k\}, \dots, \{z_L^q, z_L^k\}\}$. To ensure that z_{L-T+i}^q is as close as possible to z_{L-T+i}^k , thereby enhancing the data augmentation invariance of contrastive learning at different stages, we use the MSE loss to align these features as follows:

$$\mathcal{L}_{SPA} = \sum_{i=1}^T \|\hat{z}_{L-T+i}^k - \hat{z}_{L-T+i}^q\|_2^2, \quad (1)$$

where $\hat{z} = z/\|z\|_2$.

4.2. Prototype-based Holistic-Partial Contrastive Learning

Object parts, such as the red head and dotted tail of a bird, play a pivotal role in the characterization of fine-grained visual objects [45]. The ability to capture these discriminative parts and derive powerful part-level features is essential for accurate fine-grained self-supervised learning. In our work, we leverage the concept of prototype clustering to identify different parts of the image in the latent space and then implement fine-grained contrastive learning by combining whole-level and part-level features.

Prototype Clustering For an input sample, when it is fed into an L -stage CNN model, we can obtain an output tensor $X_L \in \mathbb{R}^{H \times W \times C}$ at the last layer, where H , W , and C denote the height, width, and number of channels, respectively.

This tensor can also be interpreted as a set of HW deep feature descriptors, denoted as $\mathbf{X}_L = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{HW}]$, where $\mathbf{x}_i \in \mathbb{R}^C$. Subsequently, to cluster these deep feature descriptors, we introduce N learnable prototypes, denoted as $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N]$, where $\mathbf{p}_i \in \mathbb{R}^C$. These prototypes are used to group the descriptors into N clusters, which are shared across all categories. Mathematically, we first normalize both \mathbf{X}_L and \mathbf{P} to calculate the cosine similarity matrix \mathbf{M} between them, as follows:

$$\mathbf{M} = \hat{\mathbf{P}}^\top \hat{\mathbf{X}}_L, \quad (2)$$

where $\hat{\mathbf{P}} = [\hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2, \dots, \hat{\mathbf{p}}_N]$ and $\hat{\mathbf{p}}_i = \mathbf{p}_i / \|\mathbf{p}_i\|_2$, similar to $\hat{\mathbf{X}}_L$. Each element m_{ij} of the matrix \mathbf{M} lies within the range $[0, 1]$ and represents the similarity between the prototype \mathbf{p}_i and the descriptor \mathbf{x}_j . This can be conceptualized as a clustering process. However, unlike traditional clustering methods, we do not assign each descriptor \mathbf{x}_j to a specific cluster to avoid an empty cluster. Instead, we use the similarities in \mathbf{M} to compute a weighted sum of the descriptors, resulting in N vectors that represent the part information.

$$\mathbf{A} = \mathbf{X}_L \mathbf{M}^\top = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N]. \quad (3)$$

At last, we average the matrix \mathbf{A} to fuse these part features for contrastive learning.

$$\mathbf{v}^{\text{part}} = \frac{\sum_{i=1}^N \mathbf{a}_i}{N}. \quad (4)$$

Holistic-Partial Contrastive Learning In addition to leverage part-level information, we also use the features produced by the last T stages after pooling to derive the whole-level feature $\mathbf{v}^{\text{whole}}$, and then concatenate the $\mathbf{v}^{\text{whole}}$ and \mathbf{v}^{part} to get \mathbf{v}^{wp} , which integrates both holistic and partial information of the image:

$$\mathbf{v}^{\text{whole}} = [\text{pool}(\mathbf{X}_{L-T+1}); \dots; \text{pool}(\mathbf{X}_L)], \quad (5)$$

$$\mathbf{v}^{\text{wp}} = [\mathbf{v}^{\text{whole}}; \mathbf{v}^{\text{part}}]. \quad (6)$$

After that, \mathbf{v}^{wp} is projected as \mathbf{q}^{wp} by the project layer proj_q or projected as \mathbf{k}^{wp} by proj_k . Then \mathbf{q}^{wp} and \mathbf{k}^{wp} are used to perform holistic-partial contrastive learning by extending the original InfoNCE loss function [14] as:

$$\mathcal{L}_{PHP} = -\log \frac{\exp(\mathbf{q}^{\text{wp}} \cdot \mathbf{k}_+^{\text{wp}} / \tau)}{\sum_{i=0}^K \exp(\mathbf{q}^{\text{wp}} \cdot \mathbf{k}_i^{\text{wp}} / \tau)}, \quad (7)$$

where τ is a temperature hyper-parameter, \mathbf{v}_q^{wp} is the feature of the query image and \mathbf{v}_k^{wp} is the feature of the key image.

4.3. Loss Function

Overall, the proposed PAPN method is end-to-end trainable by considering Eqn. (1), Eqn. (7) as:

$$\mathcal{L} = \mathcal{L}_{SPA} + \mathcal{L}_{PHP}, \quad (8)$$

where the trade-off parameters between these terms are uniformly set to 1, underscoring the non-tricky and practical nature of our method.

5. Experiments

In this section, we evaluate the performance of the propose method on four fine-grained image datasets: *Caltech UCSD-Birds (CUB200)* [40], *Stanford Cars (Cars)* [20], *FGVC-Aircraft (Aircraft)* [27], and *iNaturalist 2019(iNat2019)* [1]. Our experiments aim to understand the effectiveness and the components of our method.

5.1. Settings

Implementation Details We adopt ResNet-50 [13] as the network backbone, which comprises four stages. For a fair comparison, we follow the settings described in [38], initializing the backbone with ImageNet-1k pretrained weights for all methods listed in Table 1. Our approach is built upon MoCo v2, with the momentum value and memory size set to 0.999 and 4096, respectively. The projector head in our method consists of two fully-connected layers, each followed by ReLU activation [21] and batch normalization [18], with an output dimension of 256. The temperature parameter is set to 0.15. We set the mini-batch size as 128 and used an SGD optimizer with a learning rate of 0.03, a momentum of 0.9 and a weight decay of 0.0001. 100 epochs are used to train the feature extractor. The number of prototypes N for clustering is set to 5. We set $T = 3$ to select the last three stages for stage-wise progressive augmentation, with $\{s_1, s_2, s_3\}$ set to $\{0.45, 0.7, 0.95\}$ for all datasets, and the reduction dimension D of g is set to 128. During training, the images from the datasets are resized to 224×224 . During testing, images are first resized to 256 pixels and then are center cropped to 224×224 .

Evaluation Protocols We evaluate the proposed method in two primary settings: linear probing classification and image retrieval. Linear probing classification [30, 38] is a widely adopted evaluation protocol in SSL, where the backbone trained via SSL is fixed, and a linear classifier is subsequently trained on top of the learned features. The classification performance of this linear classifier serves as an indicator of the quality of the learned representations. Image retrieval [43] is another critical method for assessing the effectiveness of representation learning, which can also be viewed as a nearest-neighbor classification task. This task aims to find images belonging to the same category as the query images based on the learned features. We report three common metrics: top-1 accuracy for linear probing classification, rank-1 and rank-5 accuracy for image retrieval.

Table 1. Performance comparison of various methods on classification and retrieval tasks. The results include top-1 classification accuracy (in %), as well as rank-1 and rank-5 retrieval accuracies (in %). All results of baselines except DINOv2 are reported in [38, 43]. ResNet-50[†]: Indicates the use of ImageNet-1k pretraining weights, where the backbone is frozen and only the linear classifier is fine-tuned. DINOv2*: Results reproduced using the official implementation adapted for ViT-S, with weights pre-trained on ImageNet-1k for a fair comparison.

Methods	Classification			Retrieval					
	<i>CUB200</i>	<i>Cars</i>	<i>Aircraft</i>	<i>CUB200</i>		<i>Cars</i>		<i>Aircraft</i>	
				rank-1	rank-5	rank-1	rank-5	rank-1	rank-5
ResNet-50 [†]	63.06	61.41	49.79	40.39	68.94	29.28	54.66	29.48	51.39
supervised	77.46	88.60	85.93	74.43	90.99	80.54	94.35	74.16	90.47
SimSiam [6]	46.75	45.72	38.52	16.24	-	12.45	-	18.49	-
MoCo v2 [7]	63.98	62.02	51.13	39.72	67.14	30.51	56.15	30.02	52.87
LEWEL [17]	64.59	62.91	51.90	39.91	-	32.36	-	31.09	-
ContrastiveCrop [29]	64.23	63.29	52.04	39.84	-	32.71	-	30.37	-
SAM-SSL-Bilinear [37]	64.94	62.85	52.83	40.08	-	33.19	-	30.52	-
BarlowTwins [52]	33.45	31.91	34.77	-	-	-	-	-	-
VICReg [3]	37.78	30.80	36.00	-	-	-	-	-	-
LCR [38]	65.24	63.96	53.22	41.26	-	34.74	-	31.55	-
DINOv2* [28]	66.02	64.47	54.67	41.65	70.28	34.01	57.12	32.79	57.64
LDF [43]	66.17	65.60	55.28	42.06	69.59	35.81	61.94	33.27	56.80
Our PAPN	69.93	67.48	60.13	45.39	72.81	35.98	59.94	35.13	58.75

5.2. Main Results

Comparison with Previous Results. As shown in Table 1, our method achieves state-of-the-art performance across both classification and retrieval tasks on the *CUB200*, *Cars*, and *Aircraft* datasets. Notably, it outperforms the strongest baseline, LDF, in classification accuracy, with significant gains observed particularly on fine-grained distinctions. These improvements underscore the effectiveness of our approach in capturing subtle inter-class variations, which are critical for fine-grained recognition.

Our method also demonstrates robust performance in retrieval tasks, achieving consistent rank-1 and rank-5 accuracy improvements across all datasets. By leveraging a hypersphere embedding strategy, our model effectively attracts similar samples while repelling dissimilar ones in the latent space, leading to more discriminative representations. This design is particularly advantageous for handling both rigid objects (*Cars*, *Aircraft*) and non-rigid objects (*CUB200*), showcasing its versatility and practical applicability in real-world scenarios where labeled data is scarce.

Performance on the Large-Scale Dataset. We further evaluate our method’s performance on a large-scale dataset and its adaptability when the source dataset for pretraining differs from the target evaluation dataset. To this end, we pre-train our method and competitive baselines on *iNat2019* and evaluate performances on downstream fine-grained datasets. As shown in Table 2, our method consistently achieves strong results, highlighting its robustness and generalizability. These findings confirm the effectiveness of our method, even when scaling to large datasets or transferring knowledge across domains.

5.3. Ablation Studies

Key Modules We validate the proposed modules through ablation studies on the *CUB200* dataset. Table 3 summarizes the results for key components, with baseline (#1) representing MoCo v2. Introducing SPA (#2 vs. #1) significantly improves classification and retrieval accuracy, attributed to its stage-wise semantic alignment. Adding PHP (#3 vs. #1) further enhances performance by capturing whole- and part-level features for holistic-partial contrastive learning. Finally, integrating SPA and PHP (#4 vs. #1) achieves the best results, leveraging both modules for superior FGVR performance.

Intensity of Data Augmentation We investigate the impact of varying augmentation intensities defined in Section 4.1, as shown in Table 4. Initially, we set $s_2 = 0$ and $s_3 = 0$, then increase s_1 . The best performance is achieved at $s_1 = 0.45$, with further increases degrading accuracy. Next, keeping $s_1 = 0.45$ and $s_3 = 0$, we vary s_2 , finding its optimal value to be 0.70. Finally, with $s_1 = 0.45$ and $s_2 = 0.70$ fixed, we vary s_3 , determining its optimal value as 0.95. Results indicate that shallow stages (s_1) are highly sensitive to augmentation, showing significant accuracy fluctuations, while deep stages (s_3) are more robust, tolerating higher intensities with minimal accuracy loss. These findings align with the observations in Section 3.

Number of Stages To examine the impact of stage selection on model performance, we vary T , the number of final stages selected from encoders. The left part of Figure 4a shows that as T increases, the performance improve steadily, peaking at $T = 3$. However, further increasing T to 4 degrades performance, indicating that shallower stages may

Table 2. Performance of methods on the large-scale dataset. We pretrain all methods on the training set of *iNat2019* and evaluate them on different downstream fine-grained datasets. Top-1 classification accuracies (in %) are reported. The results of baselines are supplemented using the official implementations.

Methods	Source Datasets	Evaluation Datasets			
		<i>iNat2019</i> val	<i>CUB200</i> test	<i>Cars</i> test	<i>Aircraft</i> test
MoCo v2 [6]	<i>iNat2019</i> train	43.23	36.66	37.62	30.63
DINOv2 [28]	<i>iNat2019</i> train	46.90	53.14	47.42	38.73
Our PAPAN	<i>iNat2019</i> train	48.35	53.83	49.78	40.47

Table 3. Top-1 classification accuracy (in %) and rank-1, rank5 retrieval accuracy (in %) of different configs in ablation studies.

Configs	SPA	PHP	<i>CUB200</i>		
			top-1	rank-1	rank-5
#1			63.98	39.72	67.14
#2	✓		67.65	43.77	71.34
#3		✓	67.91	43.31	70.70
#4	✓	✓	69.93	45.39	72.81

Table 4. Comparisons of different values of set $\{s_1, s_2, s_3\}$. Top-1, rank-1 and rank-5 accuracies (in %) are reported.

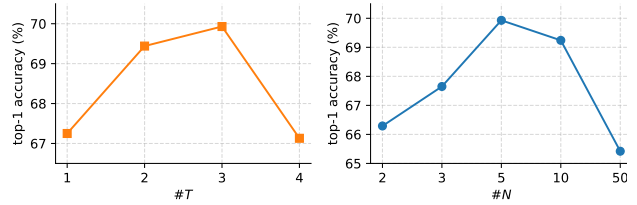
# $\{s_1, s_2, s_3\}$	<i>CUB200</i>		
	top-1	rank-1	rank-5
{0.40, 0.00, 0.00}	66.46	42.10	70.70
{0.45, 0.00, 0.00}	67.37	43.51	71.43
{0.50, 0.00, 0.00}	66.72	42.12	70.95
{0.45, 0.60, 0.00}	67.63	44.36	71.58
{0.45, 0.70, 0.00}	68.24	44.55	71.91
{0.45, 0.80, 0.00}	67.82	44.27	71.67
{0.45, 0.70, 0.80}	68.08	45.15	72.17
{0.45, 0.70, 0.95}	69.93	45.39	72.81
{0.45, 0.70, 1.00}	68.41	44.72	72.69

contain less task-relevant information.

Number of Prototypes In the PHP module, deep descriptors are grouped into N prototypes. As shown in the right part of Figure 4b, varying N impacts performance. With $N = 2$, descriptors are categorized into foreground and background, achieving 66.29% accuracy. Performance improves significantly at $N = 5$, but larger values lead to diminishing returns or even declines, likely due to over-parameterization.

5.4. Visualizing Part Localizations

To visualize part localizations from prototype clustering, we compute the similarity between each part vector \mathbf{a}_i and descriptors $[\mathbf{x}_1, \dots, \mathbf{x}_{HW}]$, averaging and normalizing these values into a heatmap in the range $[0, 1]$. The heatmap is interpolated to match the original image size. As shown in Figure 5, the model highlights specific parts (e.g., bird heads and wings, airplane engines and tails, car wheels and bodies), effectively capturing diverse part details for both rigid and non-rigid objects, demonstrating the method’s effectiveness.



(a) Impact of number of stages. (b) Impact of number of prototypes.

Figure 4. Ablation studies analyzing the effects of the number of stages and prototypes on *CUB200*. Top-1 classification accuracies (in %) are shown.

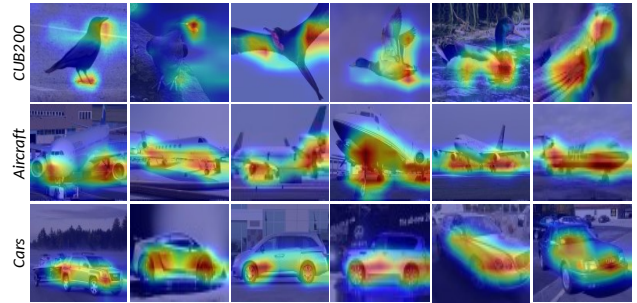


Figure 5. Visualization of part localizations on three datasets.

6. Conclusion

This study mitigated the problem of SSL for FGVR, particularly the multi-stage alignment strategy, which excels on generic datasets but struggles with fine-grained distinctions. We proposed the Progressive Augmentation Prototype-based Network (PAPAN) method for fine-grained representation learning under self-supervised settings. PAPAN introduced a progressive augmentation strategy that adapts to varying levels of augmentation tolerances at different stages, thereby maintaining semantic consistency throughout the process of stage-wise alignment. Furthermore, by leveraging prototype clustering to model object parts, PAPAN generated both global and local features for holistic-partial contrastive learning, which in turn enhances the discriminability of the learned representations. Experiments on multiple fine-grained datasets validated the effectiveness of our method and its components. Future work will explore the effectiveness of PAPAN for visual language models [35] and other downstream generative applications [19].

References

- [1] iNaturalist challenge datasets. https://github.com/visipedia/inat_comp. Accessed: 2020-11-14. 2, 6
- [2] Yuexuan An, Hui Xue, Xingyu Zhao, and Lu Zhang. Conditional self-supervised learning for few-shot classification. In *Int. Joint Conf. on Artificial Intelligence*, pages 2140–2146, 2021. 1
- [3] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 7
- [4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE Int. Conf. Comput. Vis.*, pages 9650–9660, 2021. 2
- [5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Int. Conf. Mach. Learn.*, pages 1597–1607, 2020. 1, 2
- [6] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 15750–15758, 2021. 2, 7, 8
- [7] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 2, 7
- [8] Elijah Cole, Xuan Yang, Kimberly Wilber, Oisín Mac Aodha, and Serge Belongie. When does contrastive visual representation learning work? In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 14755–14764, 2022. 2, 3
- [9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 248–255, 2009. 2
- [10] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. Technical report, California Institute of Technology, 2007. 1, 2
- [11] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent—a new approach to self-supervised learning. In *Advances in Neural Inf. Process. Syst.*, pages 21271–21284, 2020. 2
- [12] Jie Gui, Tuo Chen, Jing Zhang, Qiong Cao, Zhenan Sun, Hao Luo, and Dacheng Tao. A survey on self-supervised learning: Algorithms, applications, and future trends. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 9052–9071, 2024. 2
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 770–778, 2016. 1, 4, 6
- [14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 9729–9738, 2020. 1, 2, 6
- [15] Xiangteng He and Yuxin Peng. Fine-grained image classification via combining vision and language. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 5994–6002, 2017. 2
- [16] Feiran Hu, Chenlin Zhang, Jiangliang Guo, Xiu-Shen Wei, Lin Zhao, Anqi Xu, and Lingyan Gao. An asymmetric augmented self-supervised learning method for unsupervised fine-grained image hashing. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 17648–17657, 2024. 3, 4
- [17] Lang Huang, Shan You, Mingkai Zheng, Fei Wang, Chen Qian, and Toshihiko Yamasaki. Learning where to learn in cross-view self-supervised learning. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 14451–14460, 2022. 1, 2, 7
- [18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 6
- [19] Jian Jin, Yang Shen, Zhenyong Fu, and Jian Yang. Customized generation reimaged: Fidelity and editability harmonized. In *Eur. Conf. Comput. Vis.*, pages 410–426, 2024. 8
- [20] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *IEEE Int. Conf. Comput. Vis.*, pages 554–561, 2013. 2, 6
- [21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Inf. Process. Syst.*, pages 1097–1105, 2012. 6
- [22] Junnan Li, Pan Zhou, Caiming Xiong, and Steven C. H. Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020. 4
- [23] Zeyi Li, Pan Wang, and Zixuan Wang. FlowGANAnomaly: Flow-based anomaly network intrusion detection with adversarial learning. *Chin. J. Electron.*, pages 58–71, 2024. 1
- [24] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear CNN models for fine-grained visual recognition. In *IEEE Int. Conf. Comput. Vis.*, pages 1449–1457, 2015. 2
- [25] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear convolutional neural networks for fine-grained visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(6): 1309–1322, 2018. 2
- [26] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 3431–3440, 2015. 1
- [27] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 2, 6
- [28] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2, 7, 8
- [29] Xiangyu Peng, Kai Wang, Zheng Zhu, Mang Wang, and Yang You. Crafting better contrastive views for siamese representation learning. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 16031–16040, 2022. 7

- [30] Chau Pham, Truong Vu, and Khoi Nguyen. LP-OVOD: Open-vocabulary object detection by linear probing. In *IEEE Winter Conf. Appl. Comput. Vis.*, pages 779–788, 2024. 6
- [31] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 49–58, 2016. 1
- [32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, 2017. 1
- [33] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *IEEE Int. Conf. Comput. Vis.*, pages 618–626, 2017. 2
- [34] Yang Shen, Xuhao Sun, Xiu-Shen Wei, Qing-Yuan Jiang, and Jian Yang. SEMICON: A learning-to-hash solution for large-scale fine-grained image retrieval. In *Eur. Conf. Comput. Vis.*, pages 531–548, 2022. 1
- [35] Yang Shen, Xiu-Shen Wei, Yifan Sun, Yuxin Song, Tao Yuan, Jian Jin, Heyang Xu, Yazhou Yao, and Errui Ding. Explanatory Instructions: Towards unified vision tasks understanding and zero-shot generalization. *arXiv preprint arXiv:2412.18525*, 2024. 8
- [36] Yang Shen, Peng Wang, Xiu-Shen Wei, and Yazhou Yao. An empirical study on training paradigms for deep supervised hashing: Y. shen et al. *Int. J. Comput. Vis.*, pages 1–39, 2025. 1
- [37] Yangyang Shu, Baosheng Yu, Haiming Xu, and Lingqiao Liu. Improving fine-grained visual recognition in low data regimes via self-boosting attention mechanism. In *Eur. Conf. Comput. Vis.*, pages 449–465, 2022. 7
- [38] Yangyang Shu, Anton Van den Hengel, and Lingqiao Liu. Learning common rationale to improve self-supervised representation for fine-grained visual recognition problems. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 11392–11401, 2023. 2, 6, 7
- [39] Marcel Simon and Erik Rodner. Neural activation constellations: Unsupervised part model discovery with convolutional networks. In *IEEE Int. Conf. Comput. Vis.*, pages 1143–1151, 2015. 2
- [40] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, California Institute of Technology, 2011. 1, 2, 3, 6
- [41] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Int. Conf. Mach. Learn.*, pages 9929–9939, 2020. 2
- [42] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 3024–3033, 2021. 1
- [43] Zihu Wang, Lingqiao Liu, Scott Ricardo Figueroa Weston, Samuel Tian, and Peng Li. On learning discriminative features from synthesized data for self-supervised fine-grained visual recognition. In *Eur. Conf. Comput. Vis.*, pages 101–117, 2024. 2, 6, 7
- [44] Xiu-Shen Wei, Yang Shen, Xuhao Sun, Han-Jia Ye, and Jian Yang. A²-Net: Learning attribute-aware hash codes for large-scale fine-grained image retrieval. In *Advances in Neural Inf. Process. Syst.*, pages 5720–5730, 2021. 1
- [45] Xiu-Shen Wei, Yi-Zhe Song, Oisín Mac Aodha, Jianxin Wu, Yuxin Peng, Jinhui Tang, Jian Yang, and Serge Belongie. Fine-grained image analysis with deep learning: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(12):8927–8948, 2022. 1, 2, 5
- [46] Xiu-Shen Wei, Yu-Yan Xu, Chen-Lin Zhang, Gui-Song Xia, and Yu-Xin Peng. CAT: A coarse-to-fine attention tree for semantic change detection. *Vis. Intell.*, page 3, 2023. 1
- [47] Di Wu, Siyuan Li, Zelin Zang, Kai Wang, Lei Shang, Baigui Sun, Hao Li, and Stan Z. Li. Align yourself: Self-supervised pre-training for fine-grained recognition via saliency alignment. *arXiv preprint arXiv:2106.15788*, 2021. 2
- [48] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 3733–3742, 2018. 1
- [49] Tianjun Xiao, Yichong Xu, Kuiyuan Yang, Jiaying Zhang, Yuxin Peng, and Zheng Zhang. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 842–850, 2015. 2
- [50] Furong Xu, Meng Wang, Wei Zhang, Yuan Cheng, and Wei Chu. Discrimination-aware mechanism for fine-grained representation learning. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 813–822, 2021. 1
- [51] Ceyuan Yang, Zhirong Wu, Bolei Zhou, and Stephen Lin. Instance localization for self-supervised detection pretraining. In *IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 3987–3996, 2021. 1
- [52] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow Twins: Self-supervised learning via redundancy reduction. In *Int. Conf. Mach. Learn.*, pages 12310–12320, 2021. 7
- [53] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Eur. Conf. Comput. Vis.*, pages 818–833, 2014. 4
- [54] Chunhui Zhang, Yixiong Chen, Li Liu, Qiong Liu, and Xi Zhou. HICO: Hierarchical contrastive learning for ultrasound video model pretraining. In *Asian Conf. Comput. Vis.*, pages 229–246, 2022. 1, 2
- [55] Ruru Zhang, Haihong E, and Meina Song. FSCIL-EACA: Few-shot class-incremental learning network based on embedding augmentation and classifier adaptation for image classification. *Chin. J. Electron.*, pages 139–152, 2024. 1
- [56] Zhe Zhang, Bilin Wang, Zhezhou Yu, and Fengzhi Zhao. Attention guided enhancement network for weakly supervised semantic segmentation. *Chin. J. Electron.*, pages 896–907, 2023. 1
- [57] Xin-yang Zhao, Jian Jin, Yang-yang Li, and Yazhou Yao. Twofold debiasing enhances fine-grained learning with coarse labels. In *Proc. AAAI Conf. on Artificial Intelligence*, pages 22831–22839, 2025. 1