An Asymmetric Augmented Self-Supervised Learning Method for Unsupervised Fine-Grained Image Hashing

Feiran Hu[1] Chenlin Zhang[3] Jiangliang Guo[4] Xiu-Shen Wei[2,*]
Lin Zhao[1,*] Anqi Xu[5] Lingyan Gao[4]
[1]School of Computer Science and Engineering, Nanjing University of Science and Technology
[2]School of Computer Science and Engineering, and Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications, Southeast University
[3]4Paradigm Inc. [4]AInnovation Technology Group Co., Ltd [5]University of Toronto

Abstract

Unsupervised fine-grained image hashing aims to learn compact binary hash codes in unsupervised settings, addressing challenges posed by large-scale datasets and dependence on supervision. In this paper, we first identify a granularity gap between generic and fine-grained datasets for unsupervised hashing methods, highlighting the inadequacy of conventional self-supervised learning for fine-grained visual objects. To bridge this gap, we propose the Asymmetric Augmented Self-Supervised Learning (A^2-SSL) method, comprising three modules. The asymmetric augmented SSL module employs suitable augmentation strategies for positive/negative views, preventing fine-grained category confusion inherent in conventional SSL. Part-oriented dense contrastive learning utilizes the Fisher Vector framework to capture and model fine-grained object parts, enhancing unsupervised representations through part-level dense contrastive learning. Self-consistent hash code learning introduces a reconstruction task aligned with the self-consistency principle, guiding the model to emphasize comprehensive features, particularly fine-grained patterns. Experimental results on five benchmark datasets demonstrate the superiority of A^2-SSL over existing methods, affirming its efficacy in unsupervised fine-grained image hashing.

1. Introduction

Fine-grained image retrieval [44] in computer vision and pattern recognition aims to retrieve images from multiple subordinate categories within a super-category, *aka* a meta-category. Examples include different species of animals/plants [37], various models of cars [17], assorted types of defects [1], and diverse types of retail products [43], among others. The primary challenge lies in discerning fine-grained visual differences that subtly distinguish objects with high overall similarity but varying fine-grained features. Additionally, fine-grained retrieval necessitates the ranking of all instances so that images depicting the same sub-category label receive the highest rank, based on the fine-grained details in the query.

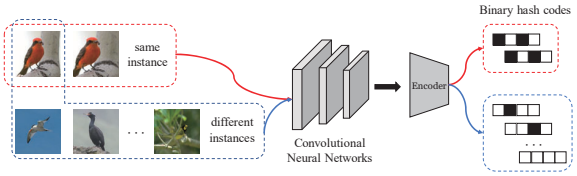

Figure 1. Illustration of the unsupervised fine-grained hashing task, aimed at generating compact binary codes with Hamming distance for fine-grained images without class labels. The goal is to ensure that images within a particular subcategory share identical codes, while images from different subcategories possess distinct codes.

Particularly, given the explosive growth of fine-grained data in real-world applications [2, 12, 21, 37, 43], fine-grained hashing emerges as a promising solution for handling large-scale fine-grained retrieval tasks. It has demonstrated the ability to significantly reduce storage costs and enhance query speeds [7, 14, 33], leveraging learned compact binary hash code representations. While prior works, *e.g.*, [7, 14, 33], achieved commendable retrieval performance, they still rely on fine-grained category annotations provided by domain experts. In some cases, obtaining extensive and accurate fine-grained supervision information is challenging or even impossible. In this study, aiming to mitigate the dependence on supervision, particularly for the task in Figure 1, we propose the development of an unsupervised

*Corresponding author. This work was supported by National Key R&D Program of China (2021YFA1001100), National Natural Science Foundation of China under Grant (62272231, 62172222), Natural Science Foundation of Jiangsu Province of China under Grant (BK20210340), the Fundamental Research Funds for the Central Universities (4009002401), and CAAI-Huawei MindSpore Open Fund.

fine-grained image hashing method, termed as Asymmetric Augmented Self-Supervised Learning (A^2-SSL).

Originally, based on preliminary experiments, we observed that on generic image datasets (*e.g.*, *NUS-WIDE* [5]), state-of-the-art methods in both unsupervised and supervised settings perform similarly. However, on fine-grained image datasets (*e.g.*, *CUB200-2011* [38]), supervised hashing methods exhibit normal accuracy, achieving approximately 80% mAP. In contrast, unsupervised hashing methods perform significantly worse than their supervised counterparts, to the extent of being ineffective (yielding less than 17% mAP). This discrepancy highlights a clear "granularity gap" between fine-grained and generic datasets when it comes to the performance of unsupervised hashing methods. Given that numerous existing unsupervised hashing methods rely on contrastive Self-Supervised Learning (SSL) [3], our hypothesis emerges from a simplification experiment involving data augmentation in self-supervision. We propose that the random augmentation commonly employed in conventional SSL may not be suitable for fine-grained visual objects.

Based on the aforementioned findings, we present the A^2-SSL method, comprising three modules: asymmetric augmented SSL, part-oriented dense contrastive learning, and self-consistent hash code learning. In asymmetric augmented SSL, we adopt a straightforward and singular data augmentation method for the anchor view of an image sample, preserving its fine-grained pattern to generate positive views. Conversely, we employ complex and diverse data augmentation methods to create negative views. Then, recognizing the significance of fine-grained object parts [44], *e.g.*, the bird's head or tail, we utilize the Fisher Vector framework [30] to capture and model these object parts in an end-to-end fashion. Leveraging these object parts, we conduct part-oriented dense contrastive learning to enhance unsupervised representations with enhanced discriminative abilities. Lastly, acknowledging that self-supervised learning primarily emphasizes overall visual similarity between image samples, potentially overlooking subtle yet discriminative fine-grained patterns, we incorporate a reconstruction task into hash learning. This addition aligns with the self-consistency principle [25], directing the model to focus on comprehensive features, particularly fine-grained patterns.

To evaluate our method, we conduct extensive experiments using five benchmark fine-grained retrieval datasets, *i.e.*, *CUB200-2011* [38], *Oxford Flowers* [27], *Stanford Dogs* [15], and *Stanford Cars* [17], *Food101* [2], for validating its effectiveness. Quantitative results of retrieval accuracy on these datasets show that the proposed A^2-SSL method obviously and consistently outperforms existing state-of-the-art methods. The ablation studies of these crucial modules in A^2-SSL also validate their own effectiveness. Furthermore, qualitative visualization results confirm that our method mitigates the limitations of existing SSL methods in fine-grained tasks.

In summary, our work has a three-fold contribution:

- We are the first to identify the granularity gap between generic datasets and fine-grained datasets for unsupervised hashing methods and address the challenge of unsupervised fine-grained image hashing.
- We propose the A^2-SSL method, comprising three crucial modules–namely, asymmetric augmented SSL, part-oriented dense contrastive learning, and self-consistent hash code learning–tailored for unsupervised fine-grained hash code learning.
- We conduct experiments on five fine-grained benchmark datasets to validate the effectiveness of A^2-SSL from both quantitative and qualitative perspectives.

2. Related Work

2.1. Large-Scale Fine-Grained Image Search

Fine-grained image search plays a pivotal role in the comprehensive analysis of fine-grained images [44]. This process involves ranking all instances in a database, where images from the same sub-category are prioritized based on fine-grained details present in the query. These techniques [9, 34] find widespread applications in real-world scenarios, such as product searches, crime prevention, and many more. SCDA [40] stands out as one of the pioneering methods employing pre-trained networks to extract and aggregate meaningful descriptors for fine-grained image search, all without explicit supervision. To enhance retrieval performance, various supervised approaches have been introduced, including learning discriminative features through centralized global pooling [52] and improving intra-class compactness with inter-class separability [53].

While these approaches have demonstrated considerable success, they grapple with the drawback of significant time consumption when searching for the nearest neighbor in extensive image databases [23]. In response to this challenge, ExchNet [7] and DSaH [14] emerged as pioneers, delving into the fine-grained hashing problem with a focus on acquiring fine-grained tailored features and generating compact binary hash codes specifically designed for fine-grained images. Subsequently, recent research has shifted its focus towards the demanding and practical task of fine-grained hashing, emphasizing the mining of discriminative regions [45] through double-filtering [4] and the creation of hash codes with strong correspondence to visual attributes [22, 33, 42]. Notably, these methods still depend on fine-grained category labels, incurring a high cost for acquisition. To our knowledge, this is the first work tailored for fine-grained hashing in an unsupervised setting.

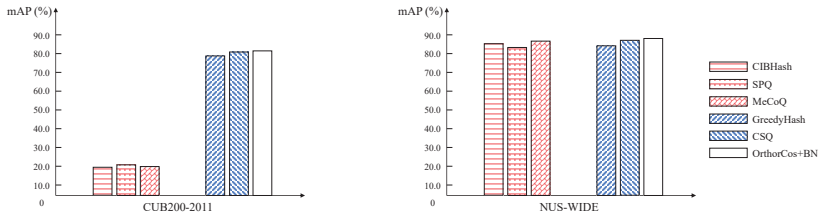

Figure 2. The granularity gap between unsupervised and supervised hashing methods on fine-grained image datasets (*e.g.*, *CUB200-2011* [38]). The retrieval accuracy (% mAP) in the figures is based on 32-bit hash codes. We also report the results of unsupervised and supervised hashing methods on generic image datasets (*e.g.*, *NUS-WIDE* [5]). Red bars represent unsupervised methods, while blue bars represent supervised methods.

2.2. Unsupervised Image Hashing

Existing deep unsupervised hashing methods primarily focus on generic images, falling into three categories [24]. The first group adopts a pseudo-label-based approach, utilizing pseudo-labels as semantic information and framing the problem as supervised hashing [50]. The second group, similarity reconstruction-based methods, employs pairwise methods to address the issue [36, 46, 47]. Due to the absence of label information, these methods typically follow a two-step framework, first extracting deep representations and then inferring similarity through distance metrics in the deep feature space. The third group embraces prediction-free self-supervised learning methods, integrating popular techniques like auto-encoders [32] and generative adversarial networks [54] into deep unsupervised hashing to extract more information through deep neural networks. Recently, leveraging the success of contrastive learning [3] in generating discriminative representations, several methods have incorporated contrastive learning into contemporary unsupervised hashing [13, 31, 39], contributing to the generation of high-quality hash codes. However, these unsupervised methods are tailored for generic images, posing challenges when applied to the task of searching images belonging to fine-grained categories.

2.3. Self-Supervised Learning

Self-supervised learning aims to acquire feature representations by minimizing a pretext task, with the supervision derived from the data itself. Examples of pretext tasks include image colorization [49], image inpainting [29], rotation prediction [8], and instance-level discrimination [3, 10]. Specifically, contrastive learning in SSL focuses on minimizing instance-level discrimination, attempting to bring embeddings of augmented views of the same image closer while pushing away embeddings from different images. While contrastive learning performs reasonably well in the unsupervised setting, it relies on overall visual similarity to group image samples, neglecting discriminative and subtle fine-grained patterns crucial for fine-grained image hashing.

3. Preliminary

Observations As is customary in evaluating the effectiveness of a proposed method, we conducted experiments comparing our unsupervised fine-grained image hashing approach with state-of-the-art methods. To do this, we tested existing unsupervised hashing methods, such as CIBHash [31], SPQ [13], and MeCoQ [39], on fine-grained benchmark datasets like *CUB200-2011* [38].

However, the empirical results of existing unsupervised hashing methods on *CUB200-2011* were less than satisfactory, with some even failing to yield meaningful outcomes. To illustrate, the retrieval accuracy, specifically mean Average Precision (mAP), for CIBHash [31] was only slightly above 15%. Similarly, other methods did not exceed 17% in accuracy. These findings raised a question: Is there ample room for improvement in the realm of unsupervised fine-grained image hashing? As a means of validation, we conducted parallel experiments involving supervised hashing methods on the *CUB200-2011* dataset, where retrieval accuracy fell within the range of 74% to 77% for methods like GreedyHash [35], CSQ [48], and OrthoCos+BN [11]. To further scrutinize this observed gap, we extended our investigations to encompass generic image hashing datasets, such as *NUS-WIDE* [5]. Strikingly, for the same methods applied in both unsupervised and supervised settings, retrieval accuracy appeared uniform. That is, when dealing with generic images, all methods from both unsupervised and supervised paradigms demonstrated comparable retrieval accuracy, as illustrated in Figure 2. This overarching observation begs the significant question: *Why does a granularity gap persist between unsupervised and supervised methods when it comes to fine-grained datasets?*

Conjecture & Discussions We have noticed that contemporary state-of-the-art unsupervised hashing methods are predominantly rooted in self-supervised learning techniques, specifically, contrastive learning [3],. Consequently, we have formulated the conjecture: *Is it possible that the existing contrastive learning methods are ill-suited for fine-grained visual objects?*

To substantiate this hypothesis, we conducted a prelimi-

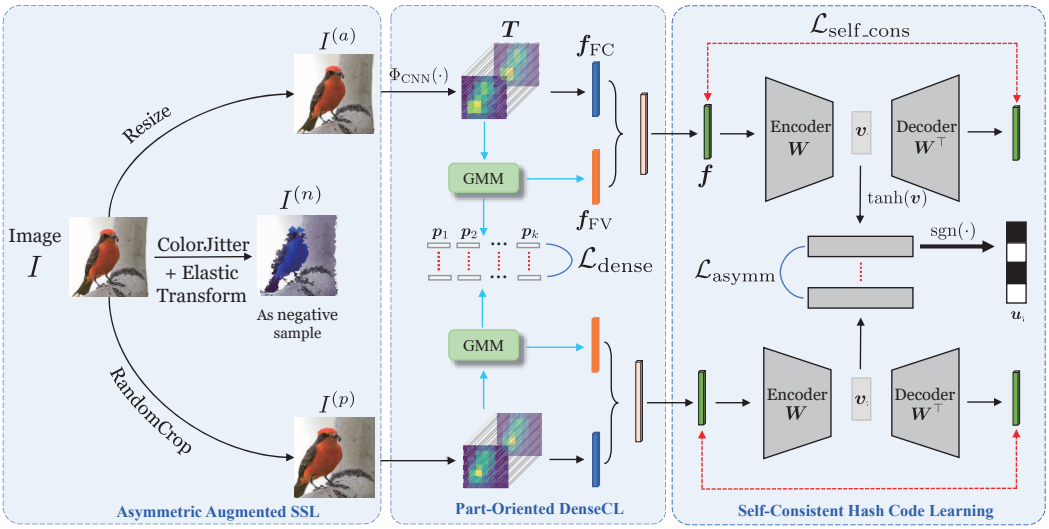

Figure 3. Overall framework of the proposed A^2-SSL method, which consists of three crucial modules, *i.e.*, asymmetric augmented SSL, part-oriented dense contrastive learning and self-consistent hash code learning. All parameters of the two branches are shared.

Table 1. Preliminary results (% mAP) of 48-bit hash codes on *CUB200-2011* in the unsupervised setting. "Traditional CL" presents contrastive learning with traditional augmentations w.r.t. the augmented views. The rest row denotes gradually removing the corresponding augmentation.

Augmentation	*CUB200-2011*
Traditional CL	15.36
w/o `GaussianBlur`	16.69
w/o `ColorJitter`	19.65
w/o `ColorJitter` & `GrayScale`	21.72
w. only `RandomCrop`	23.57

nary experiment on the *CUB200-2011* dataset, involving a gradual reduction of data augmentation within the context of contrastive learning. As depicted in Table 1, traditional contrastive learning yielded the lowest unsupervised retrieval accuracy. However, as we progressively reduced the degree of augmentation, retrieval accuracy exhibited steady improvement. Notably, the simple addition of a random crop resulted in the highest retrieval accuracy, surpassing the result achieved by traditional contrastive learning by a substantial margin (23.57% compared to 15.36%).

Therefore, the findings presented in Figure 2 and Table 1 compellingly demonstrate that the observed disparity in fine-grained visual object retrieval, often referred to as the "granularity gap", is, in fact, a consequence of self-supervised learning, specifically contrastive learning. Intriguingly, this granularity gap aligns remarkably well with the concept of coarse-grained bias introduced in [6], suggesting a connection between these phenomena.

4. Methodology

We propose an Asymmetric Augmented Self-Supervised Learning (A^2-SSL) method, which consists of three crucial modules as follows, cf. Figure 3.

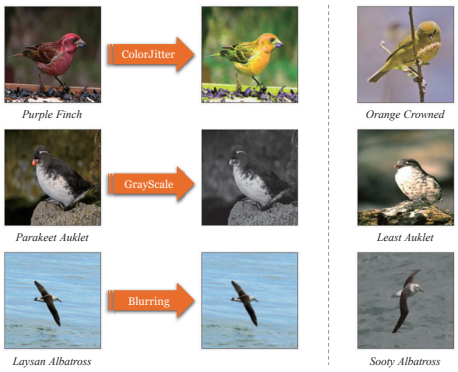

Figure 4. Several SSL augmentations (on the left) can unintentionally interfere with the inherent discriminative patterns of a specific fine-grained category, to the extent of potentially altering the category. On the right, we illustrate a case of category confusion.

4.1. Asymmetric Augmented SSL

Table 1 demonstrates that when it comes to fine-grained visual objects, employing a single and straightforward augmentation technique (such as `RandomCrop`) for self-supervised learning (SSL) yields superior results compared to using a diverse and complex set of augmentations (including `GrayScale` and `ColorJitter`, among others). This underscores the fact that not all augmentation strategies are equally suitable for fine-grained image hashing.

More specifically, traditional contrastive learning incorporates a stochastic data augmentation strategy [3], which randomly transforms any given data example to produce two related views of the same example/sample. These two views

are considered a positive pair, while other examples (usually from different categories) are treated as negative pairs. When we visualize the augmented views of fine-grained objects, we notice that certain types of augmentation can disrupt the original discriminative patterns of a specific fine-grained category to the extent of fundamentally altering that category, as depicted in Figure 4. Consequently, such augmentation methods introduce confusion in the context of self-supervised learning for fine-grained objects.

To address this challenge, we introduce a novel augmentation strategy, referred to as *asymmetric augmentation*, specifically designed to accommodate the unique characteristics of fine-grained objects within the context of SSL. In the case of a sample I, asymmetric augmentation involves creating three augmented views: the anchor view $I^{(a)}$, the positive view $I^{(p)}$, and the negative view $I^{(n)}$.

However, what sets this approach apart is the "asymmetry" in the augmentations applied to these views:

- The anchor view $I^{(a)}$ is generated solely by applying the `Resize` operation to I.
- The positive view $I^{(p)}$ is created exclusively through the `RandomCrop` operation on I.
- The negative view $I^{(n)}$ undergoes two destructive augmentations[1], namely `ColorJitter` and `ElasticTransform`, on I to induce substantial changes w.r.t. fine-grained objects.

During training, for a sample I_i ($i = (1, 2, \ldots, N)$) in a minibatch, we initially designate it as the anchor view $I_i^{(a)}$ and proceed to generate one positive view. In relation to $I_i^{(a)}$, the combination of other anchor views and positive views from the remaining $N-1$ samples, along with its original negative view $I_i^{(n)}$, results in $2(N-1)+1 = 2N-1$ negative views w.r.t. $I_i^{(a)}$. Subsequently, we directly calculate the similarity between the learned binary hash codes $\boldsymbol{u}$ for self-supervised learning, which is formulated by

$$\mathcal{L}_{\text{asymm}} = -\sum_i \log \frac{\exp\left(\boldsymbol{u}_i^{(a)} \cdot \boldsymbol{u}_i^{(p)}/\eta\right)}{r_i + \sum_{j=1}^{2N-1} \exp\left(\boldsymbol{u}_i^{(a)} \cdot \boldsymbol{u}_j^{(n)}/\eta\right)}, \tag{1}$$

where $\boldsymbol{u}_i^{(a)}$ corresponds to the hash code of $I_i^{(a)}$, $\boldsymbol{u}_i^{(p)}$ is for $I_i^{(p)}$, and $\boldsymbol{u}_j^{(n)}$ is for the negative views. $r_i = \exp\left(\boldsymbol{u}_i^{(a)} \cdot \boldsymbol{u}_i^{(p)}/\eta\right)$ is the contrastive loss for the positive view. The temperature hyper-parameter η is set to 0.3.

4.2. Part-Oriented Dense Contrastive Learning

Object parts, *e.g.*, the red head and dotted tail of a bird, play a pivotal role in the characterization of fine-grained visual objects [44]. The ability to capture these discriminative object parts and subsequently derive powerful part-level features is essential for accurate fine-grained image hashing. In our work, we advocate for the acquisition of part-oriented representations by integrating deep features into an end-to-end framework based on Fisher Vector (FV) [30]. Significantly, this approach enables us to perform dense contrastive learning using these part-level representations, facilitating more discriminative unsupervised feature learning.

Specifically, for an input sample I, by feeding it into a CNN model (without fully connected layers), we can obtain a 3D activation tensor by

$$\boldsymbol{T} = \Phi_{\text{CNN}}(I) \in \mathbb{R}^{C \times H \times W}, \tag{2}$$

where C, H and W are the depth, height, and width of the activation tensor, respectively. Alternatively, $\boldsymbol{T}$ can also be viewed as a set of $H \times W$ deep descriptors [40], denoted as, $X = \{\boldsymbol{x}_t\}$, $t = (1, 2, \ldots, H \times W)$. Subsequently, for X, we employ Gaussian Mixture Model (GMM) to cluster these deep descriptors into K clusters shared across all categories, where each cluster could correspond to a specific part-level semantic [20, 51]. Mathematically, we represent the parameters of GMM with K components/clusters by $\lambda = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k; k = 1, \ldots, K\}$, where ω_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\sigma}_k$ are the mixture weight, mean vector and covariance matrix of the k-th Gaussian component, respectively. Notably, the mixture weights ω_k are subject to the constraint:

$$\forall_k : \omega_k \geq 0, \sum\nolimits_{k=1}^{K} \omega_k = 1, \tag{3}$$

which also serves as the soft-assignment weight for the deep descriptor $\boldsymbol{x}_t$ w.r.t. the k-th cluster. Following the assignment, the mean of deep descriptors within each cluster is calculated to derive the part-level prototype:

$$\boldsymbol{p}_k = \frac{1}{|\Omega_k|} \sum_{t \in \Omega_k} \boldsymbol{x}_t, \tag{4}$$

where Ω_k represents the set of indices of deep descriptors corresponding to the k-th cluster.

After that, we perform dense contrastive learning by extending the original contrastive loss function (*i.e.*, Eqn. (1)) to a part-oriented paradigm, which can be formulated by

$$\mathcal{L}_{\text{dense}} = -\sum_i \sum_k \omega_k \log \frac{\exp\left(\boldsymbol{p}_{i,k}^{(a)} \cdot \boldsymbol{p}_{i,k}^{(p)}/\eta\right)}{\exp\left(\boldsymbol{p}_{i,k}^{(a)} \cdot \boldsymbol{p}_{i,k}^{(p)}/\eta\right) + s_{i,k}}, \tag{5}$$

where $\boldsymbol{p}_{i,k}$ denotes the k-th part-level representation of I_i, and its superscript is the anchor/positive/negative view, respectively. $s_{i,k} = \sum_{j=1}^{2N-1} \exp\left(\boldsymbol{p}_{i,k}^{(a)} \cdot \boldsymbol{p}_{j,k}^{(n)}/\eta\right)$ is the dense contrastive loss for the negative view.

Beyond the dense contrastive learning, we can also aggregate these K part-oriented clusters into a high dimensional

[1]The term "destructive" is used in the context of fine-grained objects, as these operations can potentially alter their fine-grained categories.

vector containing high-order statistics and pool them to form an image-level signature. The inclusion of high-order statistics is beneficial for modeling fine-grained patterns [19, 41]. Specifically, utilizing the mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\sigma}_k$ of the k-th Gaussian component, we can derive the first-order and second-order statistics through:

$$\begin{aligned} \boldsymbol{f}_{\boldsymbol{\mu}_k} &= \frac{1}{\sqrt{\omega_k}} \sum_t \gamma_t(k) \left(\frac{\boldsymbol{x}_t - \boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k} \right), \\ \boldsymbol{f}_{\boldsymbol{\sigma}_k} &= \frac{1}{\sqrt{2\omega_k}} \sum_t \gamma_t(k) \left[\frac{(\boldsymbol{x}_t - \boldsymbol{\mu}_k)^2}{\boldsymbol{\sigma}_k^2} - 1 \right], \end{aligned} \tag{6}$$

where $\gamma_t(k)$ represents the soft assignment, which is in detail the probability for $\boldsymbol{x}_t$ generated by the k-th Gaussian:

$$\gamma_t(k) = p(k|\boldsymbol{x}_t, \lambda) = \frac{\omega_k p_k(\boldsymbol{x}_t|\lambda)}{\sum_{j=1}^{K} \omega_j p_j(\boldsymbol{x}_t|\lambda)}, \tag{7}$$

where $p_k(\cdot)$ denotes the k-th Gaussian component in GMM.

Eventually, the final Fisher Vector $\boldsymbol{f}_{\mathrm{FV}} \in \mathbb{R}^{2CK}$ is the concatenation of both $\boldsymbol{f}_{\boldsymbol{\mu}_k}$ and $\boldsymbol{f}_{\boldsymbol{\sigma}_k}$ from all K clusters.

To achieve a holistic image representation, we pass $\boldsymbol{T}$ through two fully-connected layers to aggregate $\boldsymbol{f}_{\mathrm{FC}} \in \mathbb{R}^{d'}$, then concatenate $\boldsymbol{f}_{\mathrm{FV}}$ as

$$\boldsymbol{f} = \begin{bmatrix} \boldsymbol{f}_{\mathrm{FV}} \\ \boldsymbol{f}_{\mathrm{FC}} \end{bmatrix} \in \mathbb{R}^{2CK+d'}, \tag{8}$$

which contains both image-level information (*i.e.*, $\boldsymbol{f}_{\mathrm{FC}}$) and part-level cues (*i.e.*, $\boldsymbol{f}_{\mathrm{FV}}$).

4.3. Self-Consistent Hash Code Learning

Given that the contrastive loss in SSL has a tendency to group image samples based on overall visual similarity [6], there is a risk that the model might neglect subtle yet discriminative fine-grained patterns. To enhance the model's capability to learn comprehensive patterns, especially those that are fine-grained, we introduce a reconstruction task in the hash code learning module, guided by the principle of self-consistency [25].

The self-consistency principle [25] advocates that an autonomous intelligent system should strive for a highly self-consistent model, minimizing the internal discrepancy between observed and regenerated data from the external world. In other words, the intelligent system should be proficient at reconstructing the distribution of observed data from its compressed representation to the extent that internal distinctions become indistinguishable despite its best effort. Thus, according to the self-consistency principle, the expectation is to learn more comprehensive features, encompassing not only overall visual patterns but also valuable fine-grained patterns.

More specifically, we realize the self-consistency principle through an encoder-decoder structure applied to the holistic image embedding $\boldsymbol{f}$. In detail, $\boldsymbol{f}$ is projected into the q-dimension latent space using an encoder matrix $\boldsymbol{W} \in \mathbb{R}^{q \times (2CK+d')}$ to obtain the internal latent representation $\boldsymbol{v} \in \mathbb{R}^q$. Following the principle of self-consistency, we also aim to reconstruct the input $\boldsymbol{f}$ using a decoder $\boldsymbol{W}^\top$ as a counterpart of the encoder. This process can be mathematically formulated as:

$$\min_{\boldsymbol{W}} \|\boldsymbol{F} - \boldsymbol{W}^\top \boldsymbol{W} \boldsymbol{F}\|_F^2 \quad \text{s.t. } \boldsymbol{W}\boldsymbol{F} = \boldsymbol{V}' = \tanh(\boldsymbol{V}), \tag{9}$$

where $\boldsymbol{F} = \{\boldsymbol{f}_1; \boldsymbol{f}_2; \ldots; \boldsymbol{f}_N\} \in \mathbb{R}^{(2CK+d') \times N}$ denotes the image embeddings in a minibatch, and $\boldsymbol{V} \in \mathbb{R}^{q \times N}$ corresponds to the latent representation $\boldsymbol{v}$ in a minibatch. Directly minimizing Eqn. (9) with a hard constraint is difficult to optimize. Therefore, we relax the constraint into a soft constraint, and the learning objective can be rewritten as

$$\mathcal{L}_{\text{self\_cons}} = \left\|\boldsymbol{F} - \boldsymbol{W}^\top \boldsymbol{V}'\right\|_F^2 + \|\boldsymbol{W}\boldsymbol{F} - \boldsymbol{V}'\|_F^2 . \tag{10}$$

Finally, we can derive the q-bit binary hash code $\boldsymbol{u}$ from $\boldsymbol{v}$ by

$$\boldsymbol{u} = \operatorname{sgn}(\tanh(\boldsymbol{v})) . \tag{11}$$

4.4. Out-of-Sample Extension

Overall, the proposed A^2-SSL method is end-to-end trainable by considering Eqn. (1), Eqn. (5) and Eqn. (10) as

$$\mathcal{L} = \mathcal{L}_{\text{asymm}} + \mathcal{L}_{\text{dense}} + \mathcal{L}_{\text{self\_cons}} , \tag{12}$$

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

After training, our learned model can be applied for generating binary codes for query points including unseen query points in the training phase. Specifically, we can use the following equation to generate the binary code for I':

$$\boldsymbol{u}' = \operatorname{sgn}(\tanh(\boldsymbol{W} \cdot \boldsymbol{f}')) . \tag{13}$$

5. Experiments

5.1. Empirical Protocols and Implementations

Datasets By following ExchNet [7] and SEMICON [33], our experiments are conducted on five fine-grained benchmark datasets, *i.e.*, *CUB200-2011* [38], *Oxford Flowers* [27], *Stanford Dogs* [15], and *Stanford Cars* [17], *Food101* [2]. Dataset details can be found in the supplementary materials.

Baselines In experiments, we compare our proposed method to the following competitive baselines, *i.e.*, UnsupervisedGreedyHash (UGH) [35], MLS^{3}RUDH [36], BiHalf [18], CIBHash [31], SPQ [13], MeCoQ [39] and SDC [26]. For MLS^{3}RUDH, SPQ and MeCoQ, we utilize the open-source repository from authors and for UGH, BiHalf, CIBHash and SDC, we use the reimplementation provided by [26].

Table 2. Comparisons of retrieval accuracy (% mAP) on five fine-grained datasets.

Datasets	# bits	UGH [35]	MLS3RUDH [36]	Bihalf [18]	CIBHash [31]	SPQ [13]	MeCoQ [39]	SDC [26]	**Our A^2-SSL**
CUB200-2011	12	5.65	6.32	7.61	9.83	10.16	10.61	10.01	**20.14**
	24	8.46	8.79	10.15	12.58	13.53	13.34	14.44	**26.67**
	32	9.40	10.22	10.90	14.87	15.26	15.41	15.80	**27.77**
	48	10.41	10.95	11.35	17.05	18.01	18.80	18.13	**29.19**
Oxford Flowers	12	7.28	11.56	13.56	18.47	20.25	19.12	17.91	**34.21**
	24	10.71	14.69	17.87	26.41	27.77	26.58	23.13	**43.02**
	32	12.09	18.94	19.15	29.78	30.09	30.31	26.07	**44.82**
	48	12.38	20.35	21.15	33.04	34.89	35.44	28.41	**46.37**
Stanford Dogs	12	23.97	30.08	32.42	34.55	38.94	36.16	35.34	**54.80**
	24	35.35	40.17	44.34	46.14	48.52	46.96	47.96	**61.20**
	32	39.55	44.00	48.57	48.96	53.21	50.45	52.13	**62.70**
	48	44.10	48.22	51.88	51.67	56.55	53.76	55.56	**64.56**
Stanford Cars	12	1.86	2.01	1.96	2.56	2.92	2.71	2.73	**3.30**
	24	2.32	2.42	2.34	3.36	3.53	3.33	3.27	**4.38**
	32	2.53	2.50	2.64	3.39	4.08	3.83	3.68	**4.81**
	48	2.66	2.68	2.84	3.73	4.36	4.18	4.00	**5.28**
Food101	12	4.32	5.11	6.20	7.27	7.95	7.87	7.51	**8.48**
	24	6.33	7.45	8.25	10.62	11.22	10.95	9.50	**12.89**
	32	7.29	8.63	9.24	11.74	12.20	11.56	10.43	**14.08**
	48	8.40	9.59	10.18	12.17	13.33	13.24	11.64	**15.38**

Implementation Details We implement the proposed method based on PyTorch [28] with 4 GeForce RTX 3060 GPUs. We follow the standard evaluation protocol [31, 39], and for fair comparisons, we directly use raw image pixels resized to 224×224 as input and adopt the VGG-16 as the backbone network in experiments to extract 4096-dimension global features, *i.e.*, $\boldsymbol{f}_{\mathrm{FC}}$. Regarding $\boldsymbol{f}_{\mathrm{FV}}$, a $2CK$-dimension Fisher Vector, it is further improved by the power normalization with the fact of 0.5, followed by ℓ_2-normalization [30]. Thus, the dimension of holistic representation $\boldsymbol{f}$ is $2CK + 4096$ and then is projected by a one-layer ReLU feed-forward neural network with 4096 hidden units for the latter self-consistent hash code learning. The optimizer is Adam [16] with the weight decay as 10^{-4}, and the learning rate is set to 0.0005. The total number of training epochs is 100, and the number of batch size is 64.

5.2. Main Results

Table 2 presents the mean average precision (mAP) results of unsupervised fine-grained retrieval on these five aforementioned fine-grained benchmark datasets. For each dataset, we report the results of four lengths of hash bits, *i.e.*, 12, 24, 32, and 48, for evaluations. As shown in that table, our proposed A^2-SSL method significantly and consistently outperforms the other baseline methods on these datasets.

In particular, compared with the state-of-the-art method SDC [26], MeCoQ [39] and SPQ [13], our A^2-SSL achieves 12.23% and 9.49% improvements of 24-bit and 32-bit experiments on *CUB200-2011* and *Stanford Dogs*. Moreover, A^2-SSL obtains superior results on among small-scale fine-grained datasets, *e.g.*, *Oxford Flowers* , medium-scale fine-grained datasets, *e.g.*, *CUB200-2011*, *Stanford Dogs* and *Stanford Cars*, and large-scale fine-grained datasets, *e.g.*, *Food101*. These observations validate the effectiveness of the proposed A^2-SSL, as well as its promising practicality in real applications of unsupervised fine-grained retrieval.

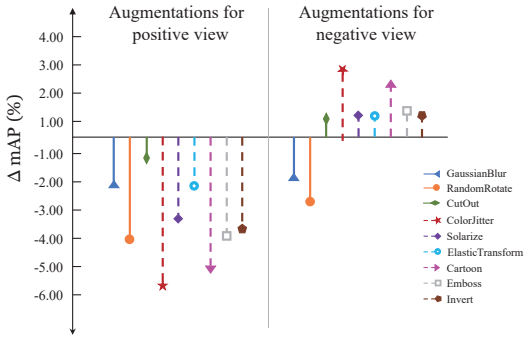

Figure 5. The mAP difference between `RandomCrop` and other augmentations for positive and negative views. For clear comparisons, the mAP results of `RandomCrop` for positive views are presented as the coordinate origin.

5.3. Ablation Studies

Key modules We validate the indispensability of A^2-SSL modules through ablation studies on the *CUB200-2011* and *Stanford Dogs* datasets. Table 3 presents ablation studies for key components, using different configurations based on the baseline (#1) that employed conventional contrastive learning in the hashing task. Specifically, a comparison between #2 and #1 reveals a substantial improvement in mAP by 11.21% and 9.16% with the introduction of asymmetric augmented SSL. The addition of part-oriented dense contrastive learning without the weight ω_k for each cluster results in a marginal 0.56% improvement (#3 vs. #2). However, incorporating the weight ω_k yields a more substantial improvement of 1.75% (cf. #5 vs. #2). Furthermore, by integrating self-consistent hash code learning, additional enhancements of

Table 3. Retrieval accuracy (% mAP) of different configurations in ablation studies of A^2-SSL.

Configurations	+ Asymmetric augmented SSL	+ Part-oriented DenseCL w/o ω_k	+ Part-oriented DenseCL w/ ω_k	+ Self-consistent	*CUB200-2011* 12-bit	*CUB200-2011* 48-bit	*Stanford Dog* 12-bit	*Stanford Dog* 48-bit
#1					8.41	15.36	30.08	47.57
#2	✓				17.62	26.57	48.16	56.93
#3	✓	✓			18.33	27.13	49.29	59.18
#4	✓	✓		✓	19.53	28.85	52.46	62.04
#5	✓		✓		19.02	28.32	51.51	61.66
#6	✓		✓	✓	20.14	29.19	54.80	64.56

Table 4. Comparisons of different numbers of clusters.

# of clusters	*CUB200-2011* 12-bit	*CUB200-2011* 48-bit
2	19.77	28.32
3	19.81	28.64
4	**20.14**	**29.19**
5	20.06	28.90
8	18.93	28.02

1.72% and 0.87% are achieved, as observed in #4 vs. #3 and #6 vs. #5, respectively.

Asymmetric augmentations To comprehensively investigate the influence of data augmentation in fine-grained hashing, we explore various common augmentation techniques. Geometric transformations, including `RandomCrop`, `RandomRotate`, and `CutOut`, represent one category of augmentation. Another category encompasses appearance transformations, such as `ColorJitter`, `Solarize`, and `ElasticTransform`.

To assess the impact of individual data augmentations on fine-grained objects, we examine the performance under setting #1 in Table 3, applying augmentations individually as per [3]. To maintain consistency, we resize all images to the same resolution due to varying sizes. The targeted transformation is then applied to generate positive or negative views, while the anchor view remains unchanged. Figure 5 illustrates the retrieval differences between the `RandomCrop` operation and others on *CUB200-2011* with a 48-bit hash code. We observe a notable decrease in performance when applying appearance transformations, such as `ColorJitter`, which may distort attributes. However, when applied to generate negative views, the retrieval accuracy increases. The most pronounced effects are observed with `ColorJitter` and `ElasticTransform`. Consequently, considering both time efficiency and accuracy enhancement, we employ them as negative augmentations, with `RandomCrop` serving as the positive one.

Number of clusters In the proposed part-oriented dense contrastive learning module, deep descriptors of each image are organized into K clusters. Table 4 presents the results for different values of K. When $K = 2$, the descriptors are roughly segmented into two parts, distinguishing between foreground and background. This baseline achieves 19.77% and 28.32% mAP. Substantial performance improvement is observed when increasing the number to 4. However, further increments in cluster number yield marginal or even negative gains, attributed to overparameterization.

Figure 6. Examples of top-10 retrieved images on *CUB200-2011* of 48-bit hash codes by our A^2-SSL.

5.4. Visualization

We visualize retrieved images on *CUB200-2011* in Figure 6, showcasing the system's proficiency in retrieving images across various subordinate categories. This capability is evident when dealing with diverse variations of the same bird species against different backgrounds. However, some failure cases are observed, particularly in instances where careful observation is needed to discern minute differences, such as those caused by variations in views between the query image and the returned images.

6. Conclusion

This study dealt with the observed "granularity gap" between fine-grained and generic datasets in the context of unsupervised hashing methods. We proposed the Asymmetric Augmented Self-Supervised Learning (A^2-SSL) method for large-scale fine-grained image retrieval under unsupervised settings. A tailored asymmetric augmentation strategy accommodated fine-grained object characteristics within the SSL framework. Leveraging the end-to-end Fisher Vector framework for modeling object parts, we employed part-oriented dense contrastive learning to enhance unsupervised representations. Integration of a reconstruction task, guided by the self-consistency principle, yielded high-quality hash codes with comprehensive features. Experiments on five fine-grained datasets validated the effectiveness of A^2-SSL and its components. Future work aims to explore the interpretability of unsupervised fine-grained hashing methods.

References

[1] Mohammad H. Alobaidi, Mohamed A. Meguid, and Tarek Zayed. Semi-supervised learning framework for oil and gas pipeline failure detection. *Scientific Reports*, 12(13758), 2022. 1

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *Proc. Eur. Conf. Comp. Vis.*, pages 446–461, 2014. 1, 2, 6

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proc. Int. Conf. Mach. Learn.*, pages 1597–1607, 2020. 2, 3, 4, 8

[4] Zhen-Duo Chen, Xin Luo, Yongxin Wang, Shanqing Guo, and Xin-Shun Xu. Fine-grained hashing with double filtering. *IEEE Trans. Image Process.*, 31:1671–1683, 2022. 2

[5] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. NUS-WIDE: A real-world web image database from national university of singapore. In *Proc. ACM Int. Conf. Image and Video Retrieval*, pages 1–9, 2009. 2, 3

[6] Elijah Cole, Xuan Yang, Kimberly Wilber, Oisin Mac Aodha, and Serge Belongie. When does contrastive visual representation learning work? In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 14755–14764, 2022. 4, 6

[7] Quan Cui, Qing-Yuan Jiang, Xiu-Shen Wei, Wu-Jun Li, and Osamu Yoshie. ExchNet: A unified hashing network for large-scale fine-grained image retrieval. In *Proc. Eur. Conf. Comp. Vis.*, pages 189–205, 2020. 1, 2, 6

[8] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *Proc. Int. Conf. Learn. Representations*, pages 1–16, 2018. 3

[9] Yangyang Guo, Zhiyong Cheng, Liqiang Nie, Xin-Shun Xu, and Mohan Kankanhalli. Multi-modal preference modeling for product search. In *Proc. ACM Int. Conf. Multimedia*, page 1865–1873, 2018. 2

[10] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, page 9729–9738, 2020. 3

[11] Jiun Tian Hoe, Kam Woh Ng, Tianyu Zhang, Chee Seng Chan, Yi-Zhe Song, and Tao Xiang. One loss for all: Deep hashing with a single cosine similarity based learning objective. In *Advances in Neural Inf. Process. Syst.*, pages 24286–24298, 2021. 3

[12] Saihui Hou, Yushan Feng, and Zilei Wang. VegFru: A domain-specific dataset for fine-grained visual categorization. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 541–549, 2017. 1

[13] Young Kyun Jang and Nam Ik Cho. Self-supervised product quantization for deep unsupervised image retrieval. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 12085–12094, 2021. 3, 6, 7

[14] Sheng Jin, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, Lei Zhang, and Xiansheng Hua. Deep saliency hashing for fine-grained retrieval. *IEEE Trans. Image Process.*, 29:5336–5351, 2020. 1, 2

[15] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. Novel dataset for fine-grained image categorization. In *CVPR Workshop on Fine-Grained Visual Categorization*, pages 806–813, 2011. 2, 6

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[17] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *ICCV Workshop on 3D Representation and Recognition*, 2013. 1, 2, 6

[18] Yunqiang Li and Jan van Gemert. Deep unsupervised image hashing by maximizing bit entropy. In *Proc. Conf. AAAI*, pages 2002–2010, 2021. 6, 7

[19] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear CNN models for fine-grained visual recognition. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 1449–1457, 2015. 6

[20] Lingqiao Liu, Chunhua Shen, and Anton van den Hengel. The treasure beneath convolutional layers: Cross-convolutional-layer pooling for image classification. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 4749–4757, 2015. 5

[21] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. DeepFashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1096–1104, 2016. 1

[22] Xin Lu, Shikun Chen, Yichao Cao, Xin Zhou, and Xiaobo Lu. Attributes grouping and mining hashing for fine-grained image retrieval. In *Proc. ACM Int. Conf. Multimedia*, page 6558–6566, 2023. 2

[23] Xin Luo, Ye Wu, and Xin-Shun Xu. Scalable supervised discrete hashing for large-scale search. In *Proc. World Wide Web Conf.*, page 1603–1612, 2018. 2

[24] Xiao Luo, Haixin Wang, Daqing Wu, Chong Chen, Minghua Deng, Jianqiang Huang, and Xian-Sheng Hua. A survey on deep hashing methods. *ACM Trans. Knowl. Discov. Data*, 17 (1), 2023. 3

[25] Yi Ma, Doris Tsao, and Heung-Yeung Shum. On the principles of parsimony and self-consistency for the emergence of intelligence. *Frontiers of Information Technology & Electronic Engineering*, 23(9):1298–1323, 2022. 2, 6

[26] Kam Woh Ng, Xiatian Zhu, Jiun Tian Hoe, Chee Seng Chan, Tianyu Zhang, Yi-Zhe Song, and Tao Xiang. Unsupervised hashing via similarity distribution calibration. *arXiv preprint arXiv:2302.07669*, 2023. 6, 7

[27] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conf. on Comput. Vision, Graph. and Image Process.*, pages 722–729, 2008. 2, 6

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Inf. Process. Syst.*, page 8024–8035, 2019. 7

[29] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, page 2536–2544, 2016. 3

[30] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *Proc. Eur. Conf. Comp. Vis.*, pages 143–156, 2010. 2, 5, 7

[31] Zexuan Qiu, Qinliang Su, Zijing Ou, Jianxing Yu, and Changyou Chen. Unsupervised hashing with contrastive information bottleneck. In *Proc. Int. Joint Conf. Artificial Intell.*, pages 959–965, 2021. 3, 6, 7

[32] Yuming Shen, Jie Qin, Jiaxin Chen, Mengyang Yu, Li Liu, Fan Zhu, Fumin Shen, and Ling Shao. Auto-encoding twin-bottleneck hashing. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2818–2827, 2020. 3

[33] 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 *Proc. Eur. Conf. Comp. Vis.*, pages 531–548, 2022. 1, 2, 6

[34] Xuemeng Song, Xianjing Han, Yunkai Li, Jingyuan Chen, Xin-Shun Xu, and Liqiang Nie. GP-BPR: Personalized Compatibility Modeling for Clothing Matching. In *Proc. ACM Int. Conf. Multimedia*, page 320–328, 2019. 2

[35] Shupeng Su, Chao Zhang, Kai Han, and Yonghong Tian. Greedy hash: Towards fast optimization for accurate hash coding in CNN. In *Advances in Neural Inf. Process. Syst.*, page 806–815, 2018. 3, 6, 7

[36] Rong-Cheng Tu, Xian-Ling Mao, and Wei Wei. MLS3RDUH: Deep unsupervised hashing via manifold based local semantic similarity structure reconstructing. In *Proc. Int. Joint Conf. Artificial Intell.*, pages 3466–3472, 2020. 3, 6, 7

[37] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The iNaturalist species classification and detection dataset. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 8769–8778, 2017. 1

[38] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD birds-200-2011 dataset. *Tech. Report CNS-TR-2011-001*, 2011. 2, 3, 6

[39] Jinpeng Wang, Ziyun Zeng, Bin Chen, Tao Dai, and Shu-Tao Xia. Contrastive quantization with code memory for unsupervised image retrieval. In *Proc. Conf. AAAI*, pages 2468–2476, 2022. 3, 6, 7

[40] Xiu-Shen Wei, Jian-Hao Luo, Jianxin Wu, and Zhi-Hua Zhou. Selective convolutional descriptor aggregation for fine-grained image retrieval. *IEEE Trans. Image Process.*, 26 (6):2868–2881, 2017. 2, 5

[41] Xiu-Shen Wei, Peng Wang, Lingqiao Liu, Chunhua Shen, and Jianxin Wu. Piecewise classifier mappings: Learning fine-grained learners for novel categories with few examples. *IEEE Trans. Image Process.*, 28(12):6116–6125, 2019. 6

[42] Xiu-Shen Wei, Yang Shen, Xuhao Sun, Han-Jia Ye, and Jian Yang. A^2-Net: Learning attribute-aware hash codes for large-scale fine-grained image retrieval. In *Advances in Neural Inf. Process. Syst.*, pages 5720–5730, 2021. 2

[43] Xiu-Shen Wei, Quan Cui, Lei Yang, Peng Wang, Lingqiao Liu, and Jian Yang. RPC: A large-scale and fine-grained retail product checkout dataset. *SCIENCE CHINA Information Sciences*, 65(9):197101, 2022. 1

[44] Xiu-Shen Wei, Yi-Zhe Song, Oisin 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

[45] Xinguang Xiang, Yajie Zhang, Lu Jin, Zechao Li, and Jinhui Tang. Sub-region localized hashing for fine-grained image retrieval. *IEEE Trans. Image Process.*, 31:314–326, 2022. 2

[46] Erkun Yang, Cheng Deng, Tongliang Liu, Wei Liu, and Dacheng Tao. Semantic structure-based unsupervised deep hashing. In *Proc. Int. Joint Conf. Artificial Intell.*, page 1064–1070, 2018. 3

[47] Erkun Yang, Tongliang Liu, Cheng Deng, Wei Liu, and Dacheng Tao. Distillhash: Unsupervised deep hashing by distilling data pairs. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2019. 3

[48] Li Yuan, Tao Wang, Xiaopeng Zhang, Francis EH Tay, Zequn Jie, Wei Liu, and Jiashi Feng. Central similarity quantization for efficient image and video retrieval. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, page 3080–3089, 2020. 3

[49] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Proc. Eur. Conf. Comp. Vis.*, pages 649–666, 2016. 3

[50] Wanqian Zhang, Dayan Wu, Yu Zhou, Bo Li, Weiping Wang, and Dan Meng. Deep unsupervised hybrid-similarity hadamard hashing. In *Proc. ACM Int. Conf. Multimedia*, page 3274–3282, 2020. 3

[51] Yu Zhang, Xiu-Shen Wei, Jianxin Wu, Cai Jianfei, Lu Jiangbo, Viet-Anh Nguyen, and Minh N. Do. Weakly supervised fine-grained categorization with part-based image representation. *IEEE Trans. Image Process.*, 25(4):1713–1725, 2016. 5

[52] Xiawu Zheng, Rongrong Ji, Xiaoshuai Sun, Yongjian Wu, Feiyue Huang, and Yanhua Yang. Centralized ranking loss with weakly supervised localization for fine-grained object retrieval. In *Proc. Int. Joint Conf. Artificial Intell.*, pages 1226–1233, 2018. 2

[53] Xiawu Zheng, Rongrong Ji, Xiaoshuai Sun, Baochang Zhang, Yongjian Wu, and Feiyue Huang. Towards optimal fine grained retrieval via decorrelated centralized loss with normalize-scale layer. In *Proc. Conf. AAAI*, pages 9291–9298, 2019. 2

[54] Maciej Zieba, Piotr Semberecki, Tarek El-Gaaly, and Tomasz Trzcinski. BinGan: Learning compact binary descriptors with a regularized gan. In *Advances in Neural Inf. Process. Syst.*, page 3608–3618, 2018. 3