

SEMICON: A Learning-to-hash Solution for Large-scale Fine-grained Image Retrieval

Yang Shen^{1,2}, Xuhao Sun¹, Xiu-Shen Wei^{1,2,3*}, Qing-Yuan Jiang¹, and Jian Yang¹

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

² State Key Laboratory of Integrated Services Networks, Xidian University, China

³ State Key Laboratory for Novel Software Technology, Nanjing University, China
{shenyang_98,sunxh,weixs,csjyang}@njust.edu.cn, qyjiang24@gmail.com

Abstract In this paper, we propose Suppression-EnhancMask based attention and Interactive Channel transformatiON (SEMICON) to learn binary hash codes for dealing with large-scale fine-grained image retrieval tasks. In SEMICON, we first develop a suppression-enhancing mask (SEM) based attention to dynamically localize discriminative image regions. More importantly, different from existing attention mechanism simply erasing previous discriminative regions, our SEM is developed to restrain such regions and then discover other complementary regions by considering the relation between activated regions in a stage-by-stage fashion. In each stage, the interactive channel transformation (ICON) module is afterwards designed to exploit correlations across channels of attended activation tensors. Since channels could generally correspond to the parts of fine-grained objects, the part correlation can be also modeled accordingly, which further improves fine-grained retrieval accuracy. Moreover, to be computational economy, ICON is realized by an efficient two-step process. Finally, the hash learning of our SEMICON consists of both global- and local-level branches for better representing fine-grained objects and then generating binary hash codes explicitly corresponding to multiple levels. Experiments on five benchmark fine-grained datasets show our superiority over competing methods. Codes are available at <https://github.com/NJUST-VIPGroup/SEMICON>.

Keywords: Fine-Grained Image Retrieval; Learning to Hash; Attention Mechanism; Large-Scale Image Search.

* Corresponding author. Y. Shen, X. Sun, X.-S. Wei and Jian Yang are also with Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, and Jiangsu Key Lab of Image and Video Understanding for Social Security, Nanjing University of Science and Technology, China. This work is supported by National Key R&D Program of China (2021YFA1001100), Natural Science Foundation of Jiangsu Province of China under Grant (BK20210340), the Fundamental Research Funds for the Central Universities (No. 30920041111, No. NJ2022028), CAAI-Huawei MindSpore Open Fund, Beijing Academy of Artificial Intelligence (BAAI), and Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX22_0463).

1 Introduction

The explosive growth of images on the web makes learning-to-hash methods become a promising solution for large-scale image retrieval tasks [44]. The objective of image-based hash learning aims to represent the content of an image by generating a binary code for both efficient storage and accurate retrieval [15]. Most existing deep hashing methods [23,15,4,17] merely support image retrieval for generic concepts, *e.g.*, cars or planes, which might fall short of practical demand with the rapidly growing amount of real applications associated with *fine-grained* image retrieval [16,40,28,32]. Thus, recent works on deep hashing [8,47,31,18] have begun to focus on fine-grained retrieval which is required to retrieve images accurately belonging to subordinate categories of a meta-category, *e.g.*, different species of animals or plants [40], rather than a generic (coarse-grained) category.

In the literature, existing generic hashing methods always utilized the outputs of the last CNN feature layer to generate binary hash codes [15,4]. Then, these generated hash codes naturally correspond to the holistic representations of the retrieved visual objects. On the other side, recent fine-grained hashing methods, some of which had achieved good retrieval accuracy, were proposed to be equipped with additional modules for locating fine-grained objects' parts (*e.g.*, birds tails or dogs heads) by region localization [31,18] or local feature alignment [8]. It is important to know that these located object parts are crucial for fine-grained vision tasks [19,42]. Eventually, similar to generic hashing methods, existing fine-grained hashing still fuses object- and part-level features as a unified feature, and then generates hash codes based on such unified features.

Therefore we ask: *What is the explicit meaning of these hash codes?* In order to make the learnt hash codes explicitly meaningful and interpretable, we propose **S**uppression-**E**nhan**C**hancing **M**ask based attention and **I**nteractive **C**hannel transformati**O**N (SEMICON), cf. Figure 1. Our SEMICON is designed by having two branches: The one is a global feature learning branch with a single global hashing unit for representing the object-level meanings, while the other one is a local pattern learning branch with multiple local hashing units for representing the multiple (different) part-level meanings in a stage-by-stage fashion. As presented in Figure 1, our final generated hash bits consists of a single object-level hash code and multiple part-level hash codes. Each hash code could explicitly correspond to its own semantic meaning.

In SEMICON, it has two crucial modules, including the suppression-enhancing mask based attention (SEM) module and the interactive channel transformation (ICON) module. More specifically, SEM is applied in each learning stage of the local pattern learning branch for dynamically localizing discriminative image regions one-by-one. However, different from other attention-based methods, our SEM is developed to restrain such regions and then discover other complementary regions by considering the relation between activated regions. Therefore, the image regions located in two adjacent stages will be correlated, which will be beneficial to the fine-grained tailored representations. For ICON, this module is employed upon each feature tensor (*e.g.*, \hat{T} in Figure 1) by adopting its channels as token embeddings to make interactions across different channels, cf. Section 3.3.

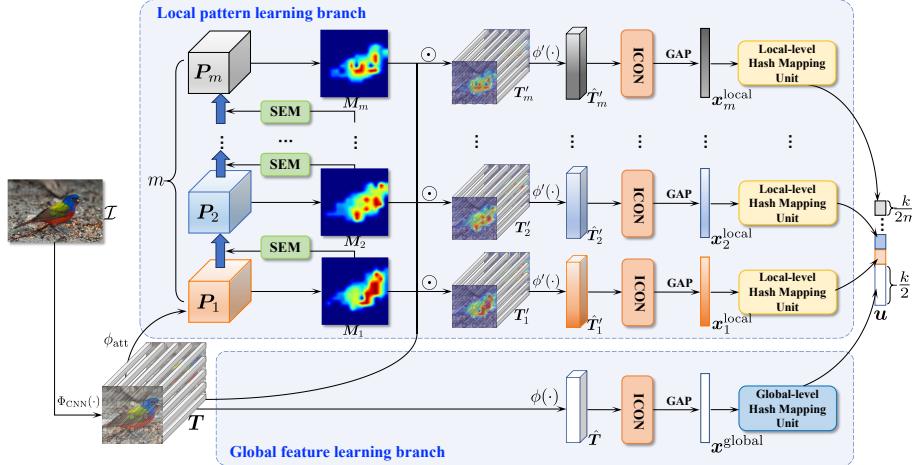


Figure 1. Overall framework of the proposed SEMICON, which consists of two branches, *i.e.*, the global feature learning branch and the local pattern learning branch. In SEMICON, the SEM module is designed to generate m attention maps (*i.e.*, \mathbf{M}_i) stage-by-stage and the ICON module takes each channel as token embeddings to make interactions among different channels. The whole network is end-to-end trainable.

Since channels can generally correspond to visual object parts [6,26,37], ICON can also model the part correlation accordingly. It could further improve fine-grained retrieval accuracy by considering the internal semantic interactions/correlations among discriminative parts [31,2]. However, as directly calculating the correlations across all channels is computationally complex, we implement this module as a two-step process in order to be efficient and scalable. Extensive experimental results on five benchmark fine-grained retrieval datasets suggest that our method achieves the new state-of-the-art performance.

The main contributions of our work are three-fold. (1) We propose a novel method, *i.e.*, the suppression-enhancing mask based attention and interactive channel transformation, for dealing with the fine-grained hash learning task. (2) We design a suppression-enhancing mask based attention operation to maintain relations between different activated regions and propose a two-step interactive channel transformation module to build correlations between different channels. (3) Experimental results on five benchmark datasets show that our SEMICON achieves significant improvements over competing methods.

2 Related Work

2.1 Fine-Grained Image Retrieval

Fine-grained retrieval is a fundamental topic of fine-grained image analysis [47] which has gained more and more traction in recent years [34,8,55,46,38]. Compared

with generic image retrieval, which focuses on retrieving similar images based on similarities in their content (*e.g.*, texture, color, and shape), fine-grained retrieval aims to retrieve the images of the same category type (*e.g.*, the same subordinate species of animals [40]) with only subtle differences (*e.g.*, different beak colors or claw shapes of birds).

Depending on the types of query images, fine-grained image retrieval tasks can be separated into two groups, *i.e.*, fine-grained content based image retrieval (FG-CBIR) and fine-grained sketch-based image retrieval (FG-SBIR). In concretely, SCDA [45] is one of the earliest work of FG-CBIR that used deep pre-trained networks without using explicit localization supervisions. Supervised metric learning based approaches were then proposed to overcome the retrieval accuracy limitations of unsupervised retrieval [2]. In the other research line, FG-SBIR is another interesting task related to both fine-grained image retrieval and cross-modal retrieval of which goal is to match specific photo instances using a free-hand sketch as the query modality. Existing FG-SBIR approaches generally aim to train embedding space where sketches and photos can be compared in a nearest neighbor fashion [49,38].

As all these fine-grained retrieval methods utilize the outputs of the last feature layer of deep networks to deal with retrieval tasks, they still have limitations in the face of large-scale data even if they have achieved good results. To be specific, the searching time for exact nearest neighbor is typically expensive or even impossible for the given queries in large-scale retrieval tasks. To alleviate this issue, fine-grained hashing, which aims to generate compact binary codes to represent fine-grained objects, as a promising direction has attracted the attention in the fine-grained community very recently [8,18,52,46].

2.2 Learning to Hash

Hashing has been widely-studied to transform the data item to a short code consisting of a sequence of bits (*i.e.*, hash codes). Compared to data-independent hashing [9,30,36], data-dependent hashing (*aka* learning to hash) aims to learn hash codes that are more compact yet more data-specific. Due to the discrete of hash codes and non-differentiability of binary hash functions, the optimization of learning to hash is NP-hard [15].

Specifically, data-independent hashing methods attempted to adjust hash generating from different perspectives, *e.g.*, the theory or machine learning views, to name a few: proposing random hash functions satisfying local sensitive property [9], developing better search schemes [30], providing faster computation of hash functions [36], etc. In contrast with data-independent hashing methods, since data-dependent hashing methods learn hash functions from a specific dataset to achieve similarity preserving, they can generally obtain superior retrieval accuracy. Especially for capitalizing on advances in deep learning, many well-performing methods were proposed to integrate feature learning and hash code learning into an end-to-end framework based on deep networks, *e.g.*, [17,3,51].

Fine-grained hashing, as a more challenging and practical hashing task in the vision community, has achieved great attention in very recent years [8,18,46,52,29].

In the literature, ExchNet [8] and DSaH [18] defined the fine-grained hashing task almost at the same time. While, they added additional modules to extract local-level features for representing objects' parts, and then aggregated both global-level features and local-level features together to generate the unified binary hash codes. SwinFGHash [29] did not add extra modules but took transformer-based architecture to model the feature interactions. The learnt hash bits of these methods seem incomprehensible and lack semantics which are meaningful to fine-grained objects as we do not know what these hash bits explicitly indicate. Although A²-NET [46] tried to equip those learnt unified hash codes with correspondence to object attributes [11], the hash mapping component still mixes up multiple levels of features together which made the hash codes ambiguous w.r.t. clear visual semantics. In this paper, we do not aggregate all the features from different levels together to generate the unified hash codes, but generate the final hash codes corresponding to the features from different levels in a stage-by-stage fashion.

2.3 Attention Mechanism

Attention mechanisms are those methods for diverting attention to the most important regions of an image and disregarding irrelevant regions [13]. In the past years, attention mechanism has played an increasingly important role and has provided benefits in many vision tasks, *e.g.*, image classification [48], image retrieval [33] and object detection [5]. In a vision system and Deep Neural Networks (DNNs), an attention mechanism can be viewed as a step of dynamically selecting and adaptively weighting features according to the importance of inputs.

Attention mechanisms can be categorised according to data domain [13]. Besides temporal attention [22] and branch attention [24], most of the existing attention mechanisms are related to channel information. In DNNs, different channels in different feature maps usually represent different objects' parts [6,26,37]. Channel attention adaptively recalibrates the weight of each channel in DNNs which can be viewed as an object selection process [13]. In fine-grained tasks, researchers often adopt the erasing operation [53,25] on the most discriminative regions, which can also be described as the most activated channels, to mine discerning information from the rest of the channels. However, these erasing based attention methods seem less informative that the relations across different regions are completely lost.

Recently, self-attention, which has achieved great success in Natural Language Processing [41], has also shown the potential to become a dominant tool in vision tasks [10,27]. Typically, self-attention is used as a spatial attention mechanism to capture global information. Nowadays, the standard Vision Transformer usually split input images into equal-sized blocks and utilize these blocks as the token embeddings [10]. To capture fine-grained parts' correlations, we propose the interactive channel transformation (ICON) module in our SEMICON and utilize different channels as token embeddings. We further implement this module as a two-step computation process in order to reduce the computational complexity.

3 Methodology

3.1 Overall Framework and Notations

Generally, both object-level (global-level) and part-level (local-level) features are crucial in fine-grained visual tasks [47]. Therefore, the overall framework of our SEMICON maintains a global feature learning branch and a local pattern learning branch, cf. Figure 1. Correspondingly, our hash code learning component consists of two units, *i.e.*, the global-level hash mapping unit and the local-level hash mapping unit. In particular, the global-level hash mapping unit is designed to capture object-level binary codes while the local-level hash mapping unit is additionally divided into m sub linear encoder paradigms, which is beneficial to obtaining part-level binary hash codes explicitly in a stage-by-stage fashion. Thus, the final learnt hash codes contain both object-level and part-level meanings. Furthermore, our proposed suppression-enhancing mask based attention (SEM) module and interactive channel transformation (ICON) module are developed to generate both discriminative global-level features and correlated local-level features.

In concretely, for each input image \mathcal{I} , a backbone CNN model $\Phi_{\text{CNN}}(\cdot)$ is used to extract its deep activation tensor \mathbf{T} :

$$\mathbf{T} = \Phi_{\text{CNN}}(\mathcal{I}) \in \mathbb{R}^{C \times H \times W}. \quad (1)$$

Then, based on \mathbf{T} , a global-level transforming network $\phi(\cdot)$, which is equipped with a stack of convolution layers, is performed within the global feature learning branch as:

$$\hat{\mathbf{T}} = \phi(\mathbf{T}; \theta_{\text{global}}) \in \mathbb{R}^{C' \times H' \times W'}, \quad (2)$$

where θ_{global} presents the parameters of $\phi(\cdot)$. The local pattern learning branch contains an attention guidance $\mathbf{P}_1 \in \mathbb{R}^{c \times H \times W}$, which is utilized to generate the attention map \mathbf{M}_1 in the first stage, cf. Section 3.2. With the help of the attention map, we can evaluate the attended deep descriptors in these $H \times W$ cells by conducting element-wise Hadamard product by:

$$\mathbf{T}'_1 = \mathbf{M}_1 \odot \mathbf{T}. \quad (3)$$

Then, the proposed SEM module is adopted to generate other attention maps \mathbf{M}_i in the following $m - 1$ stages, as well as the corresponding deep activation tensors \mathbf{T}'_i . Besides, to obtain semantic-specific representations, a local-level transforming network $\phi'(\cdot)$, which has the same structure as $\phi(\cdot)$, is used to transform \mathbf{T}'_i as

$$\hat{\mathbf{T}}'_i = \phi'(\mathbf{T}'_i; \theta_{\text{local}}) \in \mathbb{R}^{C' \times H' \times W'}, \quad (4)$$

where θ_{local} presents the parameters of $\phi'(\cdot)$. Then, the proposed ICON module is conducted over $\hat{\mathbf{T}}$ and $\hat{\mathbf{T}}'_i$ for making interactions across different channels.

Finally, by performing global average-pooling on $\hat{\mathbf{T}}$ and $\hat{\mathbf{T}}'_i$, we can obtain the object-level feature $\mathbf{x}^{\text{global}}$ and m part-level features $\mathbf{x}_i^{\text{local}}$. In order to generate the binary-like codes, a binary-like code mapping module consists of

$m + 1$ linear encoder paradigms $\mathbf{W} = \{\mathbf{W}^{\text{global}}, \mathbf{W}_1^{\text{local}}, \mathbf{W}_2^{\text{local}}, \dots, \mathbf{W}_m^{\text{local}}\}$ is built to project $\mathbf{x}^{\text{global}}/\mathbf{x}_i^{\text{local}}$ as $\mathbf{v}^{\text{global}}/\mathbf{v}_i^{\text{local}}$. Eventually, the hash code learning module is performed upon $\mathbf{v}^{\text{global}}$ and $\mathbf{v}_i^{\text{local}}$ to obtain the final binary hash codes $\mathbf{u} = [\mathbf{u}^{\text{global}}, \mathbf{u}_1^{\text{local}}, \mathbf{u}_2^{\text{local}}, \dots, \mathbf{u}_m^{\text{local}}]$.

3.2 Suppression-Enhancing Mask based Attention

Attention in human perception renders that humans selectively focus on several salient parts of an object, which may help better capture visual structure [20]. Inspired by this, we incorporate the attention mechanism into the local pattern learning branch to capture the patterns of fine-grained objects' parts.

In previous fine-grained vision tasks, some works adopt the mask based attention for erasing the most discriminative regions to mine the rest of the object-specific regions in different branches [53,25]. However, the simple erasing of the most discriminative regions seems trivial and will overlook the relations between the erased regions and other significant regions. To overcome such an issue, we propose the suppression-enhancing mask based attention (SEM) module to maintain relations among different activated regions. It is worth mentioning that the proposed SEM can be realized by convolutional layers sharing parameters, which could bring computational economy.

In concretely, for the given deep activation tensor \mathbf{T} related to the input image \mathcal{I} , m attention maps $\mathcal{M} = \{\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_m\}$ whose $\mathbf{M}_i \in \mathbb{R}^{H \times W}$ will be extracted. While the m attention guidances \mathbf{P}_i which are utilized to calculate the attention maps can be expressed as:

$$\mathbf{P}_i = \begin{cases} \phi_{\text{att}}(\mathbf{T}; \theta_{\text{att}}), & i = 1 \\ f_{\text{SEM}}(\text{softmax}(\mathbf{M}_{i-1})) \odot \mathbf{P}_{i-1}, & i = \{2, 3, \dots, m\} \end{cases}, \quad (5)$$

where ϕ_{att} is a transformation network which can be optimized in an end-to-end manner driven by the overall loss function described in Section 3.4 and f_{SEM} is the suppression-enhancing mask based attention operation which will be described later in this section.

More specifically, the initial attention map \mathbf{M}_1 is generated according to the attention guidance \mathbf{P}_1 w.r.t. \mathbf{T} in the first stage while in the following $m - 1$ stages, attention maps are generated by the suppression-enhancing mask based attention operation. To obtain \mathbf{M}_1 , a transformation network ϕ_{att} is primarily used to obtain what to pay attention to, which can be formulated as:

$$\mathbf{P}_1 = \phi_{\text{att}}(\mathbf{T}; \theta_{\text{att}}), \quad (6)$$

where $\mathbf{P}_1 \in \mathbb{R}^{c \times H \times W}$ presents the attention guidance within the first stage and θ_{att} presents the parameters of the corresponding network w.r.t. \mathbf{T} . Then, a 1×1 convolution layer φ_1 followed by ϕ_{att} is designed to gain \mathbf{M}_1 .

For the remaining attention maps $\mathbf{M}_i, i = \{2, 3, \dots, m\}$ in the following $m - 1$ stages, we perform the suppression-enhancing mask based attention operation f_{SEM} which not only helps suppress (rather than simply erasing) the previous most discriminative region but also enhance the other activated regions.

In details, we first calculate the weight of each cell in the attention map \mathbf{M}_{i-1} of the previous stage by conducting a softmax function:

$$\mathbf{M}'_{i-1} = \text{softmax}(\mathbf{M}_{i-1}) \in \mathbb{R}^{H \times W}. \quad (7)$$

Then, we record μ_{i-1}^{std} and μ_{i-1}^{mean} as the standard deviation value and the mean value of all the elements in \mathbf{M}'_{i-1} . For each element $\mu_{i-1}^k \in \{\mu_{i-1}^1, \mu_{i-1}^2, \dots, \mu_{i-1}^{H \times W}\}$ in \mathbf{M}'_{i-1} , the f_{SEM} operation is defined as follows:

$$\mu_{i-1}^k = 1 - \frac{\mu_{i-1}^k - \mu_{i-1}^{\text{mean}}}{(\mu_{i-1}^{\text{std}})^\alpha}, \quad (8)$$

where α is a hyper-parameter used to regularize the degree of suppression ratio of discriminative regions and the enhance ratio of other activated regions. Additionally, the attention guidance \mathbf{P}_{i-1} of the previous stage is then changing to \mathbf{P}_i by performing element-wise Hadamard product. The i th attention map is afterwards generated by the i th 1×1 convolution layer φ_i . Therefore, the representations of m attention maps \mathbf{M}_i can be written as:

$$\mathbf{M}_i = \varphi_i(\mathbf{P}_i), i = \{1, 2, \dots, m\}. \quad (9)$$

Thus, the final m deep activation tensors \mathbf{T}'_i can be obtained via

$$\mathbf{T}'_i = \mathbf{M}_i \odot \mathbf{T}, i = \{1, 2, \dots, m\}. \quad (10)$$

By performing this suppression-enhancing mask based attention operation, the most discriminative region in the attention guidance of the previous stage will be partially restrained. Meanwhile, those unactivated regions will be further inhibited while other activated regions will be enhanced with attention. Therefore, relations between the activated regions of the previous stage and the activated regions generated afterwards could be maintained.

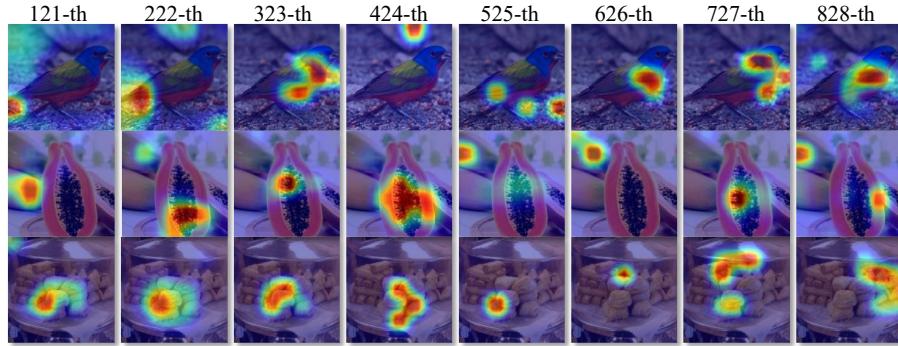


Figure 2. Visualization of channels extracted from DNNs by highlighting their weights.

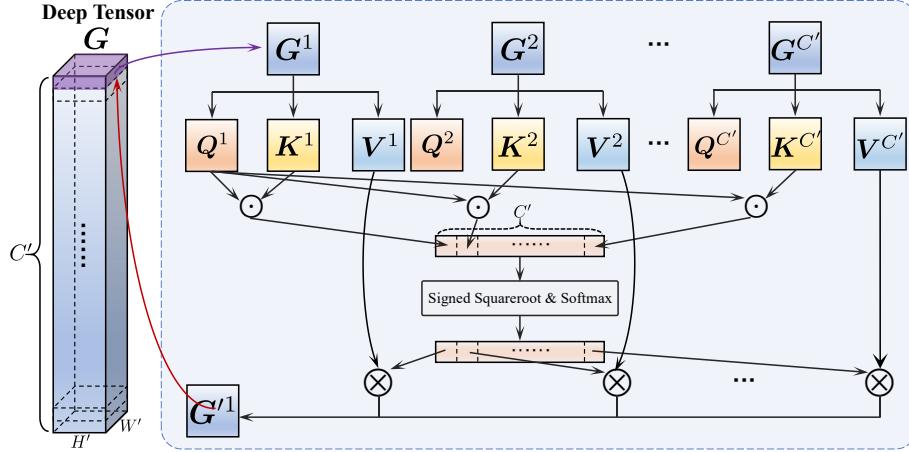


Figure 3. The Interactive Channel transformatiON (ICON) module. It utilizes each channel as token embeddings and makes interactions across different channels.

3.3 Interactive Channel Transformation

In Deep Neural Networks (DNNs), channels are usually exploited as objects’ part detectors [37,6,26]. As can be seen from Figure 2, the activated regions of the sampled channels (highlighted in warm colors) are semantically meaningful. Therefore, we incorporate the self-attention mechanism into our model and utilize each channel as token embeddings to make interactions across different channels for capturing the correlations of fine-grained “parts”, which has been proved can be greatly improved the fine-grained recognition accuracy [47,54,50,7].

In Figure 3, an overview of the proposed interactive channel transformation (ICON) module is depicted. The computational complexity of directly performing the interactive channel transformation over all channels is considerable. Therefore, for the given deep tensor $\mathbf{G} \in \{\hat{\mathbf{T}}, \hat{\mathbf{T}}'_1, \hat{\mathbf{T}}'_2, \dots, \hat{\mathbf{T}}'_m\}$ of each input image \mathcal{I} , we split it into several portions and design a two-step interactive channel transformation module (cf. Figure 4) which can be directly adopted in traditional deep hashing frameworks to reduce the computational consumption.

Specifically, the first step is composed of a stack of N identical parts. For each given \mathbf{G} , we split the deep tensor into N equal length portions $[\mathbf{G}_1; \mathbf{G}_2; \mathbf{G}_3; \dots; \mathbf{G}_N]$, where $\mathbf{G}_i \in \mathbb{R}^{d \times H' \times W'}$ and $d = C'/N$. (H', W') is the resolution of each channel while C' is the number of channels. For each \mathbf{G}_i , the interactive channel transform operation is used to generate the transformed portion \mathbf{G}'_i in order to make interactions over different channels within itself. The interactive channel transform operation during the first step can be described as mapping a unique query (\mathbf{Q}_i) and key-value ($\mathbf{K}_i - \mathbf{V}_i$) pair to an output ($\hat{\mathbf{G}}_i$), where $\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i$ are generated from \mathbf{G}_i via a 1×1 convolution layer. By following [41], we first compute the dot products $\hat{\mathbf{G}}_i$ of the query \mathbf{Q}_i with the key \mathbf{K}_i and divide by

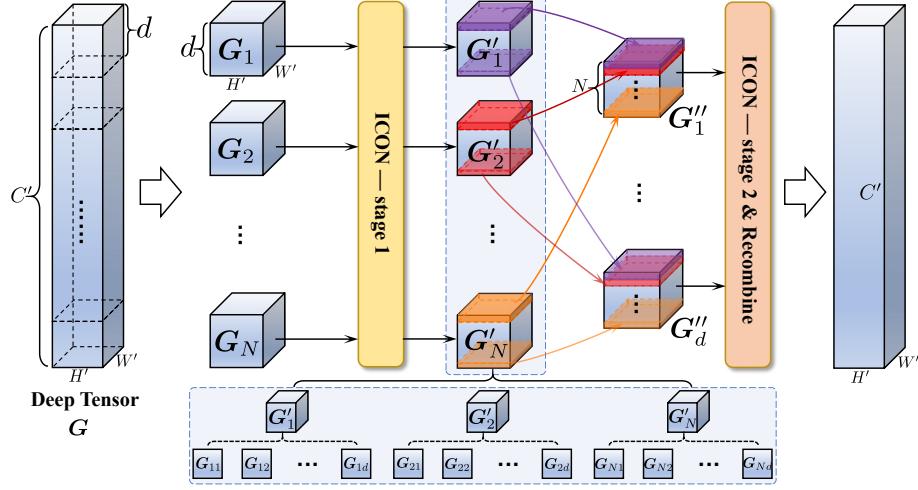


Figure 4. Our interactive channel transformation module is implemented by a two-step process for reducing the computational consumption.

\sqrt{d} :

$$\hat{\mathbf{G}}_i = \frac{\mathbf{Q}_i \mathbf{K}_i^\top}{\sqrt{d}}. \quad (11)$$

Then, a signed squareroot step and a softmax function is applied to generate each output \mathbf{G}'_i as:

$$\mathbf{G}'_i = \text{softmax} \left(\text{sign}(\hat{\mathbf{G}}_i) \cdot \sqrt{|\hat{\mathbf{G}}_i| + \delta} \right) \mathbf{V}_i, \quad (12)$$

where δ is a fixed positive bias.

In order to make interaction among different portions, in the second step, the tokens at the same position in \mathbf{G}'_i are recombined into \mathbf{G}''_i . In simpler terms, for each portion $\mathbf{G}'_i = \{\mathbf{G}_{i1}; \mathbf{G}_{i2}; \dots; \mathbf{G}_{id}\}$, where $\mathbf{G}'_i \in \mathbb{R}^{d \times H' \times W'}$ and $\mathbf{G}_{ij} \in \mathbb{R}^{H' \times W'}$ obtained from the first step, we recombine these portions by integrating those channels with the same index in preparation for the second step interactive channel transformation. To be specific, the recombined portion \mathbf{G}''_i is consisted of N channels from the previous N portions \mathbf{G}'_i :

$$\mathbf{G}''_i = \{\mathbf{G}_{1i}; \mathbf{G}_{2i}; \dots; \mathbf{G}_{Ni}\}, i = \{1, 2, \dots, d\}. \quad (13)$$

The second step ICON is then performed on \mathbf{G}''_i with the same processes as the first step. Finally, channels which have changed their original index will be reset after performing the two-step ICON process.

Between these two steps, we employ a batch normalization and a ReLU activation. A residual connection [14] is adopted after each step. Instead of performing a single interactive channel transform operation associated with keys,

values and queries, inspired by [41], we perform the two-step interactive channel transform operation in parallel. For traditional deep hashing frameworks generally use CNNs as vanilla backbones, we perform group convolutions as a substitute for multi-head linear projections. The group number across the first step is N while it will be reset as d within the second step. This two-step multi-group interactive channel transformation allows the model to jointly process information within different indexes over different channels.

3.4 Hash Code Learning

In the following, we conduct the hash code learning based on the obtained object-level features and part-level features. Assuming that we have q query data points which are denoted as $\{\mathbf{q}_i\}_{i=1}^q$, as well as p database points which are denoted as $\{\mathbf{p}_j\}_{j=1}^p$. For each \mathbf{q}_i and \mathbf{p}_j , it consists of a global feature $\mathbf{v}^{\text{global}}$ and m local features $\mathbf{v}_i^{\text{local}}$. The corresponding hash codes can be carried out by

$$\mathbf{u}_i = \text{sign}(\mathbf{q}_i), \quad \mathbf{z}_j = \text{sign}(\mathbf{p}_j), \quad (14)$$

where $\mathbf{u}_i, \mathbf{z}_j \in \{-1, +1\}^k$, and k presents the length of the final binary hash codes. The goal of hashing is to learn binary hash codes for both query points and database points and preserving their similarity simultaneously. Following [17], the formulation of the hash code learning can be written as:

$$\min_{\mathbf{W}, \Theta} \mathcal{L}(\mathcal{I}) = \sum_{i \in \Omega} \sum_{j \in \Gamma} \left[\text{sign}(\mathbf{W} \cdot F(\mathcal{I}_i; \Theta))^{\top} \mathbf{z}_j - kS_{ij} \right]^2, \quad \mathbf{z}_j \in \{-1, +1\}^k, \quad (15)$$

where Γ presents the indices of all the database points while $\Omega \subseteq \Gamma$ presents the indices of the query set points for we can only gain access to a set of database points $\{\mathbf{p}_j\}_{j=1}^p$ without query points during the training stage. $S \in \{-1, +1\}^{q \times p}$ denotes the pairwise supervised information. \mathbf{W} presents the matrix of $m+1$ linear projection and Θ denotes the parameters of DNNs to be learned.

By relaxation, we get the final formulation of SEMICON:

$$\begin{aligned} \min_{\mathbf{W}, \Theta} \mathcal{L}(\mathcal{I}) = & \beta \sum_{i \in \Omega} \sum_{j \in \Gamma} \left[\tanh(\mathbf{W} \cdot F(\mathcal{I}_i; \Theta))^{\top} \mathbf{z}_j - kS_{ij} \right]^2 \\ & + \gamma \sum_{i \in \Omega} [\mathbf{z}_i - \tanh(\mathbf{W} \cdot F(\mathcal{I}_i; \Theta))]^2, \end{aligned} \quad (16)$$

where β and γ are hyper-parameters as the trade-off. The proposed SEMICON is an end-to-end deep hashing method which is able to simultaneously perform feature learning and hash code learning in such a unified framework.

4 Experiments

4.1 Datasets

By following A²-NET [46] and ExchNet [8], our experiments are conducted on two widely used fine-grained datasets, *i.e.*, *CUB200-2011* [43] and *Aircraft* [32], as well

as three popular large-scale fine-grained datasets, *i.e.*, *Food101* [1], *NABirds* [39] and *VegFru* [16]. Specifically, *CUB200-2011* contains 11,788 bird images from 200 bird species and is officially split into 5,994 images for training and 5,794 images for test. *Aircraft* contains 10,000 images of 100 aircraft variants, among which 6,667 images for training and 3,333 images for test. For large-scale datasets, *Food101* contains 101 kinds of foods with 101,000 images, where for each class, 250 test images are checked manually for correctness while 750 training images still contain a certain amount of noises. *NABirds* contains 48,562 images of North American birds with 555 sub-categories, 23,929 images for training while 24,633 images for test. *VegFru* is another large-scale fine-grained dataset covering 200 kinds of vegetables and 92 kinds of fruits with 29,200 for training, 14,600 for validation and 116,931 for test.

4.2 Baselines and Implementation Details

Baselines In experiments, we compare our proposed method to the following competitive generic hashing methods, *i.e.*, ITQ [12], SDH [35], DPSH [23], HashNet [4], and ADSH [17]. Among them, DPSH, HashNet and ADSH are also deep learning based methods, while ITQ and SDH are not. Furthermore, we also compare the results of our SEMICON with state-of-the-arts of fine-grained hashing methods, including ExchNet [8] and A²-NET [46].

Implementation Details For fair comparisons, we follow the training setting in A²-NET [46] and ExchNet [8]. In concretely, for *CUB200-2011*, *Aircraft* and *Food101*, we only sample 2,000 images per epoch for training, while 4,000 samples are randomly selected per epoch for *NABirds* and *VegFru*. For the training details, regarding the backbone model, we can choose any network structures as the base network for fine-grained representation learning. While, by following ExchNet [8] and A²-NET [46], ResNet-50 [14] is employed in experiments for fair comparisons. The attention generation network ϕ_{att} is the fourth stage of ResNet-50 without downsample convolutions. The global-level transforming network $\phi(\cdot)$ and the local-level transforming network $\phi'(\cdot)$ are independent networks, sharing the same architecture with the fourth stage of ResNet-50. The total number of training epochs is 30. The iteration time is 40 for those datasets containing less than 20,000 training images while for other datasets, the iteration time is 50. For all datasets, we preprocess all images to 224×224 , and the learning rate is set to 2.5×10^{-4} for all iterations. SGD with mini-batch set as 16 is used for training. We set the weight decay as 10^{-4} and momentum as 0.91. The hyper-parameters, *i.e.*, α in Eq. (8) and β , γ in Eq. (16), are set as 0.3, 1 and 200, respectively. By following ADSH [17], we adopt soft-constraints strategy [21] to avoid the similarity imbalance problem. The number of m is set as 3 which means there exists 3 attention maps M_i . The length of the final hash code $\mathbf{u}^{\text{global}}$ and $\mathbf{u}_i^{\text{local}}$ is set as $\lceil \frac{k}{2} \rceil$ and $\lfloor \frac{k}{6} \rfloor$. The fixed positive bias δ is set as 10^{-5} . All experiments are conducted with one GeForce RTX 2080 Ti GPU.

Table 1. Comparisons of retrieval accuracy (% mAP) on five fine-grained datasets.

Datasets	# bits	ITQ	SDH	DPSH	HashNet	ADSH	ExchNet	A^2 -NET	Ours
<i>CUB200-2011</i>	12	6.80	10.52	8.68	12.03	20.03	25.14	33.83	37.76
	24	9.42	16.95	12.51	17.77	50.33	58.98	61.01	65.41
	32	11.19	20.43	12.74	19.93	61.68	67.74	71.61	72.61
	48	12.45	22.23	15.58	22.13	65.43	71.05	77.33	79.67
<i>Aircraft</i>	12	4.38	4.89	8.74	14.91	15.54	33.27	42.72	49.87
	24	5.28	6.36	10.87	17.75	23.09	45.83	63.66	75.08
	32	5.82	6.90	13.54	19.42	30.37	51.83	72.51	80.45
	48	6.05	7.65	13.94	20.32	50.65	59.05	81.37	84.23
<i>Food101</i>	12	6.46	10.21	11.82	24.42	35.64	45.63	46.44	50.00
	24	8.20	11.44	13.05	34.48	40.93	55.48	66.87	76.57
	32	9.70	13.36	16.41	35.90	42.89	56.39	74.27	80.19
	48	10.07	15.55	20.06	39.65	48.81	64.19	82.13	82.44
<i>NABirds</i>	12	2.53	3.10	2.17	2.34	2.53	5.22	8.20	8.12
	24	4.22	6.72	4.08	3.29	8.23	15.69	19.15	19.44
	32	5.38	8.86	3.61	4.52	14.71	21.94	24.41	28.26
	48	6.10	10.38	3.20	4.97	25.34	34.81	35.64	41.15
<i>VegFru</i>	12	3.05	5.92	6.33	3.70	8.24	23.55	25.52	30.32
	24	5.51	11.55	9.05	6.24	24.90	35.93	44.73	58.45
	32	7.48	14.55	10.28	7.83	36.53	48.27	52.75	69.92
	48	8.74	16.45	9.11	10.29	55.15	69.30	69.77	79.77

4.3 Main Results

Table 1 presents the mean average precision (mAP) results of fine-grained retrieval for comparisons with state-of-the-art hashing methods on these five aforementioned benchmark fine-grained datasets. For each dataset, we report the results of four lengths of hash bits, *i.e.*, 12, 24, 32, and 48, for evaluations. From table 1, we can observe that the proposed SEMICON significantly outperforms the other baseline methods on these datasets.

In particular, compared with the state-of-the-art method A^2 -NET [46], our SEMICON achieves 11.42% and 17.17% improvements over A^2 -NET of 24-bit and 32-bit experiments on *Aircraft* and *VegFru*. Moreover, SEMICON obtains superior results on both medium-scale fine-grained datasets, *e.g.*, *CUB200-2011* and *Aircraft*, and large-scale fine-grained datasets, *e.g.*, *NABirds* and *VegFru*. These observations validate the effectiveness of the proposed SEMICON, as well as its promising practicality in real applications of fine-grained retrieval.

4.4 Ablation Studies

We demonstrate the effectiveness of these crucial modules of the proposed SEMICON, *i.e.*, the novel hash learning framework (cf. Section 3.1), the suppression-

Table 2. Retrieval accuracy (% mAP) with incremental modules of the proposed SEMICON.

Configurations	CUB200-2011				Aircraft				Food101			
	12	24	32	48	12	24	32	48	12	24	32	48
Vanilla backbone	20.03	50.33	61.68	65.43	15.54	23.09	30.37	50.65	35.64	40.93	42.89	48.81
+ SEMICON ^{-*}	34.93	58.73	64.71	75.66	34.18	70.14	76.50	80.23	40.59	72.75	78.98	80.15
+ SEM	36.58	64.19	71.58	79.17	43.36	73.39	80.64	83.99	44.95	75.44	80.07	82.40
+ ICON	37.76	65.41	72.33	79.62	49.87	75.08	80.45	84.23	50.00	76.57	80.19	82.44

* SEMICON⁻ represents the model generates m attention maps without performing SEM and the proposed ICON is not performed before obtaining the final hash codes.

enhancing mask based attention (SEM) module (cf. Section 3.2) and the interactive channel transformation (ICON) module (cf. Section 3.3). In the ablation studies, we apply these modules incrementally on a vanilla backbone (*i.e.*, ResNet-50) as the baseline. As evaluated in Table 2, by stacking these modules one by one, the retrieval results are steadily improved, which justifies the effectiveness of our proposals in SEMICON.

5 Conclusion

In this paper, we proposed the Suppression-Enhancing Mask based Attention and Interactive Channel Transformation (SEMICON) for dealing with the large-scale fine-grained image retrieval task. In concretely, the SEM module was developed to restrain (rather than simply erasing) the most discriminative region under the attention guidance of the previous stage, which benefited maintaining relations between different activated regions in a stage-by-stage fashion. Moreover, as channels in DNNs could often correspond to object parts, our ICON module treated each channel as token embeddings for capturing fine-grained parts’ correlations. With the hash mapping component containing two units of both global-level and local-level, the final learnt binary hash codes can be generated from different features with different levels (*i.e.*, global-level and local-level) respectively. Experiments on five fine-grained datasets demonstrated the effectiveness of our SEMICON, as well as its proposals. In the future, we would like to improve the robustness of hashing methods and conduct experiments under a more generalized retrieval setting where training classes and test classes have no overlap.

Acknowledgements

The authors would like to thank the anonymous reviewers for their critical and constructive comments and suggestions. We gratefully acknowledge the support of MindSpore, CANN (Compute Architecture for Neural Networks) and Ascend AI Processor used for this research.

References

1. Bossard, L., Guillaumin, M., Gool, L.V.: Food-101 – mining discriminative components with random forests. In: Proc. Eur. Conf. Comp. Vis. pp. 446–461 (2014)
2. Cai, S., Zuo, W., Zhang, L.: Higher-order integration of hierarchical convolutional activations for fine-grained visual categorization. In: Proc. IEEE Int. Conf. Comp. Vis. pp. 511–520 (2017)
3. Cakir, F., He, K., Sclaroff, S.: Hashing with binary matrix pursuit. In: Proc. Eur. Conf. Comp. Vis. pp. 332–348 (2018)
4. Cao, Z., Long, M., Wang, J., Yu, P.S.: HashNet: Deep learning to hash by continuation. In: Proc. IEEE Int. Conf. Comp. Vis. pp. 5608–5617 (2017)
5. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Proc. Eur. Conf. Comp. Vis. pp. 213–229 (2020)
6. Chen, L., Zhang, H., Xiao, J., Nie, L., Shao, J., Liu, W., Chua, T.S.: SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. pp. 5659–5667 (2017)
7. Chen, Y., Bai, Y., Zhang, W., Mei, T.: Destruction and construction learning for fine-grained image recognition. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. pp. 5157–5166 (2019)
8. Cui, Q., Jiang, Q.Y., Wei, X.S., Li, W.J., Yoshie, O.: ExchNet: A unified hashing network for large-scale fine-grained image retrieval. In: Proc. Eur. Conf. Comp. Vis. pp. 189–205 (2020)
9. Dasgupta, A., Kumar, R., Sarlos, T.: Fast locality-sensitive hashing. In: Proc. ACM SIGKDD Int. Conf. Knowledge discovery & data mining. pp. 1073–1081 (2011)
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
11. Ferrari, V., Zisserman, A.: Learning visual attributes. In: Proc. Advances in Neural Inf. Process. Syst. pp. 433–440 (2007)
12. Gong, Y., Lazebnik, S., Gordo, A., Perronnin, F.: Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. IEEE Trans. Pattern Anal. Mach. Intell. **35**(12), 2916–2929 (2012)
13. Guo, M.H., Xu, T.X., Liu, J.J., Liu, Z.N., Jiang, P.T., Mu, T.J., Zhang, S.H., Martin, R.R., Cheng, M.M., Hu, S.M.: Attention mechanisms in computer vision: A survey. arXiv preprint arXiv:2111.07624 (2021)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. pp. 770–778 (2016)
15. Hoe, J.T., Ng, K.W., Zhang, T., Chan, C.S., Song, Y.Z., Xiang, T.: One loss for all: Deep hashing with a single cosine similarity based learning objective. In: Proc. Advances in Neural Inf. Process. Syst. (2021)
16. Hou, S., Feng, Y., Wang, Z.: VegFru: A domain-specific dataset for fine-grained visual categorization. In: Proc. IEEE Int. Conf. Comp. Vis. pp. 541–549 (2017)
17. Jiang, Q.Y., Li, W.J.: Asymmetric deep supervised hashing. In: Proc. Conf. AAAI. pp. 3342–3349 (2018)
18. Jin, S., Yao, H., Sun, X., Zhou, S., Zhang, L., Hua, X.: Deep saliency hashing for fine-grained retrieval. IEEE Trans. Image Process. **29**, 5336–5351 (2020)
19. Krause, J., Gebru, T., Deng, J., Li, L.J., Fei-Fei, L.: Learning features and parts for fine-grained recognition. In: Proc. Int. Conf. Patt. Recogn. pp. 26–33 (2014)

20. Larochelle, H., Hinton, G.: Learning to combine foveal glimpses with a third-order boltzmann machine. In: Proc. Advances in Neural Inf. Process. Syst. pp. 1243–1251 (2010)
21. Leng, C., Cheng, J., Wu, J., Zhang, X., Lu, H.: Supervised hashing with soft constraints. In: Proc. ACM Int. Conf. Information & Knowledge Management. pp. 1851–1854 (2014)
22. Li, J., Wang, J., Tian, Q., Gao, W., Zhang, S.: Global-local temporal representations for video person re-identification. In: Proc. IEEE Int. Conf. Comp. Vis. pp. 3958–3967 (2019)
23. Li, W.J., Wang, S., Kang, W.C.: Feature learning based deep supervised hashing with pairwise labels. In: Proc. Int. Joint Conf. Artificial Intell. pp. 1711–1717 (2015)
24. Li, X., Wang, W., Hu, X., Yang, J.: Selective kernel networks. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. pp. 510–519 (2019)
25. Liu, C., Xie, H., Zha, Z., Yu, L., Chen, Z., Zhang, Y.: Bidirectional attention-recognition model for fine-grained object classification. *IEEE Trans. Multimedia* **22**(7), 1785–1795 (2019)
26. Liu, L., Shen, C., Van den Hengel, A.: The treasure beneath convolutional layers: Cross-convolutional-layer pooling for image classification. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. pp. 4749–4757 (2015)
27. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proc. IEEE Int. Conf. Comp. Vis. pp. 10012–10022 (2021)
28. Liu, Z., Luo, P., Qiu, S., Wang, X., Tang, X.: DeepFashion: Powering robust clothes recognition and retrieval with rich annotations. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. pp. 1096–1104 (2016)
29. Lu, D., Wang, J., Zeng, Z., Chen, B., Wu, S., Xia, S.T.: SwinFGHash: Fine-grained image retrieval via transformer-based hashing network. In: Proc. British Machine Vis. Conf. pp. 1–13 (2021)
30. Lv, Q., Josephson, W., Wang, Z., Charikar, M., Li, K.: Multi-probe LSH: Efficient indexing for high-dimensional similarity search. In: Proc. Int. Conf. Very Large Data Bases. pp. 950–961 (2007)
31. Ma, L., Li, X., Shi, Y., Wu, J., Zhang, Y.: Correlation filtering-based hashing for fine-grained image retrieval. *IEEE Signal Processing Letters* **27**, 2129–2133 (2020)
32. Maji, S., Rahtu, E., Kannala, J., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. arXiv preprint arXiv:1306.5151 (2013)
33. Ng, T., Balntas, V., Tian, Y., Mikolajczyk, K.: SOLAR: second-order loss and attention for image retrieval. In: Proc. Eur. Conf. Comp. Vis. pp. 253–270 (2020)
34. Pang, K., Yang, Y., Hospedales, T.M., Xiang, T., Song, Y.Z.: Solving mixed-modal jigsaw puzzle for fine-grained sketch-based image retrieval. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. pp. 10347–10355 (2020)
35. Shen, F., Shen, C., Liu, W., Shen, H.T.: Supervised discrete hashing. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. pp. 37–45 (2015)
36. Shrivastava, A., Li, P.: Densifying one permutation hashing via rotation for fast near neighbor search. In: Proc. Int. Conf. Mach. Learn. pp. 557–565 (2014)
37. Simon, M., Rodner, E.: Neural activation constellations: Unsupervised part model discovery with convolutional networks. In: Proc. IEEE Int. Conf. Comp. Vis. pp. 1143–1151 (2015)
38. Song, J., Yu, Q., Song, Y.Z., Xiang, T., Hospedales, T.M.: Deep spatial-semantic attention for fine-grained sketch-based image retrieval. In: Proc. IEEE Int. Conf. Comp. Vis. pp. 5551–5560 (2017)

39. Van Horn, G., Branson, S., Farrell, R., Haber, S., Barry, J., Ipeirotis, P., Perona, P., Belongie, S.: Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. pp. 595–604 (2015)
40. Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., Belongie, S.: The iNaturalist species classification and detection dataset. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. pp. 8769–8778 (2018)
41. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Proc. Advances in Neural Inf. Process. Syst. pp. 5998–6008 (2017)
42. Vedaldi, A., Mahendran, S., Tsogkas, S., Maji, S., Girshick, R., Kannala, J., Rahtu, E., Kokkinos, I., Blaschko, M.B., Weiss, D.: Understanding objects in detail with fine-grained attributes. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. pp. 3622–3629 (2014)
43. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD birds-200-2011 dataset. Tech. Report CNS-TR-2011-001 (2011)
44. Wang, J., Zhang, T., Sebe, N., Tao, S.H.: A survey on learning to hash. IEEE Trans. Pattern Anal. Mach. Intell. **40**(4), 769–790 (2017)
45. Wei, X.S., Luo, J.H., Wu, J., Zhou, Z.H.: Selective convolutional descriptor aggregation for fine-grained image retrieval. IEEE Trans. Image Process. **26**(6), 2868–2881 (2017)
46. Wei, X.S., Shen, Y., Sun, X., Ye, H.J., Yang, J.: A²-NET: Learning attribute-aware hash codes for large-scale fine-grained image retrieval. In: Proc. Advances in Neural Inf. Process. Syst. pp. 5720–5730 (2021)
47. Wei, X.S., Song, Y.Z., Mac Aodha, O., Wu, J., Peng, Y., Tang, J., Yang, J., Belongie, S.: Fine-grained image analysis with deep learning: A survey. IEEE Trans. Pattern Anal. Mach. Intell. (2021), doi:10.1109/TPAMI.2021.3126648
48. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: CBAM: Convolutional block attention module. In: Proc. Eur. Conf. Comp. Vis. pp. 3–19 (2018)
49. Yu, Q., Liu, F., Song, Y.Z., Xiang, T., Hospedales, T.M., Loy, C.C.: Sketch me that shoe. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. pp. 799–807 (2016)
50. Yu, Y., Tang, S., Aizawa, K., Aizawa, A.: Category-based deep cca for fine-grained venue discovery from multimodal data. IEEE Trans. Neural Netw. & Learn. Syst. **30**(4), 1250–1258 (2018)
51. Yuan, X., Ren, L., Lu, J., Zhou, J.: Relaxation-free deep hashing via policy gradient. In: Proc. Eur. Conf. Comp. Vis. pp. 134–150 (2018)
52. Zeng, Z., Wang, J., Chen, B., Dai, T., Xia, S.T.: Pyramid hybrid pooling quantization for efficient fine-grained image retrieval. arXiv preprint arXiv:2109.05206 (2021)
53. Zhang, X., Wei, Y., Feng, J., Yang, Y., Huang, T.S.: Adversarial complementary learning for weakly supervised object localization. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. pp. 1325–1334 (2018)
54. Zheng, H., Fu, J., Mei, T., Luo, J.: Learning multi-attention convolutional neural network for fine-grained image recognition. In: Proc. IEEE Int. Conf. Comp. Vis. pp. 5209–5217 (2017)
55. Zheng, X., Ji, R., Sun, X., Zhang, B., Wu, Y., Huang, F.: Towards optimal fine grained retrieval via decorrelated centralized loss with normalize-scale layer. In: Proc. Conf. AAAI. pp. 9291–9298 (2019)