Self-supervised Label-Visual Correlation Hashing for Multi-label Image Retrieval

Yu Liu[1], Yanzhao Xie[1], Jingkuan Song[2(✉)], Rukai Wei[1], and Ke Zhou[1]

[1] Huazhong University of Science and Technology, Wuhan, China
{liu_yu,yzxie,weirukai,zhke}@hust.edu.cn
[2] University of Electronic Science and Technology of China, Chengdu, China
jingkuan.song@gmail.com

Abstract. Perceiving multiple objects within an image without the labels' supervision is the challenge of multi-label image hashing tasks. Existing unsupervised hashing approaches do reconstruction or contrastive learning for the representation of the object of interest but ignore the other objects in the image. We propose to use pseudo labels to provide candidate objects, making the image match the possible objects' features by the co-occurrence correlations between labels. As a result, we explore the co-occurrence correlations based on empirical models and design a data augmentation strategy in a self-supervised learning framework to learn label-level embeddings. We also build the image visual correlations and design a dual overlapping group sum-pooling (OGSP) component to fuse label-level and visual-level embeddings into image representations, alleviating noise from empirical models. Extensive experiments on public multi-label image datasets using pseudo labels demonstrate that our self-supervised label-visual correlation hashing framework outperforms state-of-the-art label-free hashing algorithms for retrieval. GitHub address: https://github.com/lzHZWZ/SS-LVH.git.

Keywords: Multi-label image hashing · Self-supervised learning · Co-occurrence correlations

1 Introduction

In label-free scenarios, image hashing algorithms [25] remain tricky for learning accurate hash codes for an image containing multiple objects. Existing unsupervised hashing methods ignore the existence of other objects and only perceive the object of interest in an image, resulting in limited performance. If features of all objects are extracted in advance using techniques like object detection [9], the computational cost will be a huge problem.

Recently, the study of co-occurrence correlation [4] has attracted our interest. This correlation reveals the probability of different objects appearing in an image. It can serve as potential supervisory information to aid in the perception of objects of interest. Meanwhile, since the co-occurrence correlation reflects a

B. Li et al. (Eds.): APWeb-WAIM 2022, LNCS 13422, pp. 129–143, 2023.
https://doi.org/10.1007/978-3-031-25198-6_10

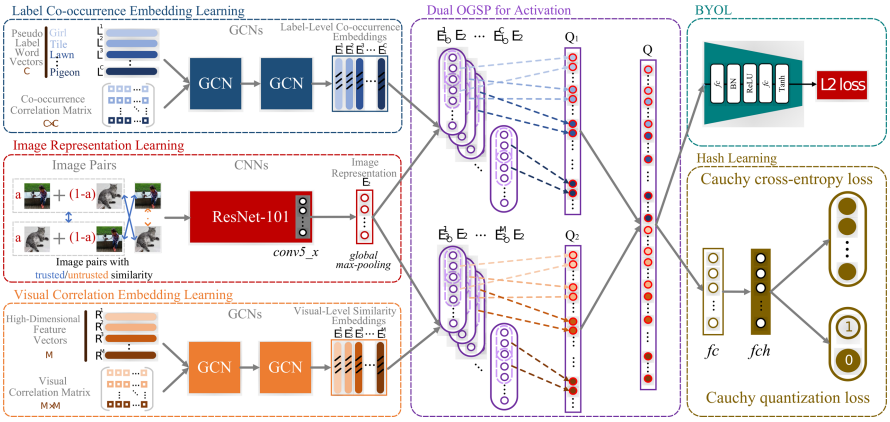


Fig. 1. The architecture of SS-LVH. (1) In the label co-occurrence embedding learning branch (blue frame), C denotes the number of labels, L^1 to L^C denote word vectors corresponding to pseudo labels, and E_1^1 to E_1^C denote label-level co-occurrence embeddings corresponding to L^1 to L^C. (2) In the image representation learning branch (red frame), the input is image pairs with trusted or untrusted similarity. *Conv5_x* is a layer generating the image feature. E_2 denotes the image representation calculated via *global max-pooling* layer with the feature. (3) In the visual correlation embedding learning branch (orange frame), M denotes the number of images sampled from the target dataset, R^1 to R^M denote high-dimensional feature vectors corresponding to sampled images, and E_3^1 to E_3^M denote visual-level similarity embeddings corresponding to R^1 to R^M. (4) The purple frame denotes the dual OGSP component. Based on *overlapping group (a dotted box) sum-pooling*, Q_1 (*i.e.*, semantic similarity representations) is fused by each E_1^i ($i \in \{1, 2, \ldots, C\}$) and E_2, and Q_2 (*i.e.*, visual similarity representations) is fused by each E_3^j ($j \in \{1, 2, \ldots, M\}$) and E_2, where $\circ$ denotes Hadmard Product. The label-visual representation Q is acquired by concatenating Q_1 with Q_2. (5) The cyan frame completes self-supervised learning for Q in the way of BYOL. The *Tanh* function is added to improve the adaptation of hashing. (6) The golden frame achieves hash learning by the Cauchy distribution loss functions [25] consisting of Cauchy cross-entropy loss and Cauchy quantization loss. (Color figure online)

common phenomenon in the real world, it is reliable in label-free scenarios. As a result, the labels' co-occurrence correlations of the empirical model can provide relatively accurate priori information. Although some labels in the empirical model may not exist in the target dataset, co-occurrence correlation can ensure that the relevant objects (*e.g.*, basketball and players) in the image are simultaneously activated by Graph Convolutional Networks (GCNs) [25], partially solving the multi-object perception problem of unlabeled images.

Based on this motivation, we propose to incorporate co-occurrence correlations of pseudo labels [16] (*i.e.*, labels of the empirical model) into a self-supervised learning (SSL) framework [8] to design a multi-object hashing model. The architecture is shown in Fig. 1. We gather the co-occurrence probability of each pseudo label to build the adjacency matrix, and input the matrix into GCNs for the label-level embedding learning. To alleviate the noise caused by applying the empirical models to out-of-distribution (o.o.d) data, we introduce

the adjacency matrix based on the visual correlations of all/sampling images (See Sect. 3), and input the matrix into the other GCN branch for the visual-level embedding learning. Since it is derived from images rather than labels, this visual-level embedding is more representative of the distribution of the target dataset [2]. We also use a feature extraction backbone to generate image representations. With the embeddings and image representations, we design a dual *overlapping group sum-pooling* (OGSP) component to fuse them. The embeddings and representations are fused into a vector by Hadmard Product and then is mapped to multiple cells by *group sum-pooling* with overlapped windows. Compared with the Multi-modal Factorized Bilinear (MFB) component used in LAH [25], the dual OGSP component preserves richer spatial information. As a result, the regions of interest will be highlighted through more representations. Furthermore, it can balance the activated representations based on two embeddings, improving generalization ability and accuracy. Finally, we employ the Cauchy distribution loss functions [1] to learn the activated representations into hash codes.

In this paper, we propose a self-supervised label-visual correlation hashing (SS-LVH) framework for multi-label image retrieval. In practice, we employ Bootstrap Your Own Latent (BYOL) [8] as the SSL framework in that we can learn compact representations without negative sampling. For this limitation, we design a data augmentation strategy that fuses the two images via different weights as the pretext task, used to enhance the learning for co-occurrence correlations. In addition, we incorporate the *Tanh* function into the BYOL framework to adapt hash learning. For the embeddings learning, we use BERT [5] to generate label embeddings and select the Classification Transformer (C-Tran) model pre-trained on Visual Genome 500 (VG-500) [13]) as the empirical model, the ResNet-101 [24] model as the representation backbone. Note that we will first use the BYOL framework to pre-train the model, and then access the hash loss functions for hashing. Extensive experiments on public multi-label image datasets using pseudo labels demonstrate that SS-LVH is conducive to retrieving images that share at least one label. Its performance is better than state-of-the-art label-free hashing methods. In addition, we demonstrate that all the components we introduced can improve retrieval performance.

The contributions of SS-LVH are summarized below.

(1) We proposed a novel SSL framework, *i.e.*, SS-LVH, for image hashing using co-occurrence correlations of pseudo labels. By perceiving multiple objects in an image via pseudo labels and their co-occurrence correlations, we achieve self-supervised hashing in a multi-label learning pattern.
(2) We designed a series of tricks to resist noise from o.o.d data, including the label-visual correlation learning scheme and the dual OGSP component, resulting in accurate multi-object activation.
(3) SS-LVH outperforms state-of-the-art unsupervised/self-supervised hashing methods in terms of multi-label image retrieval on three public datasets. We demonstrate that the co-occurrence correlations can benefit the label-free hash learning performance.

2 Related Work

The SS-LVH is designed via the co-occurrence information of objects in images and the SSL framework. Recent research in these fields is described below.

Co-occurrence Correlations. The object co-occurrence correlations in the images can represent the intrinsic logical relation of objects included by images. Wang *et al.* propose CNN-RNN [22], utilizing the semantic redundancy and the co-occurrence dependency to construct an end-to-end classification model. ML-GCN [4] is a novel trainable multi-object image recognition framework, which employs GCN to map the label representations (*i.e.*, word embeddings), including co-occurrence information and inter-dependency of objects in images. In SS-LVH, we also exploit this insight to construct a co-occurrence correlation matrix to delegate the object's inter-dependency.

Multi-label Image Hashing. The multi-label hashing methods can improve the accuracy of image retrieval. Lai *et al.* propose Instance-aware hashing (IAH) [12], which first conducts the instance-aware retrieval via learning-based hashing. Song *et al.* propose Deep Region Hashing (DRH) [20] with a cost-free hashing strategy, and can generate the hash codes for whole image as well as the object candidate regions. Xie *et al.* propose Label-Attended Hashing (LAH) [25] that combines the co-occurrence correlations of labels to learn hash codes.

Self-supervised Learning. We follow SSL for guiding our model to acquire the appropriate image representations without hand-crafted labels. In this field, contrastive methods [3,8] have shown impressive results, with the fundamental ideology pulling representations of different views transformed from the same sample closer together (*i.e.*, positive pairs) while spreading representations of different data views (*i.e.*, negative pairs). Chen *et al.* propose the method SimCLR [3] based on contrastive insight, which utilizes a learnable nonlinear transformation between data representations and the contrastive loss, thus improving the quality of representations. BYOL [8] utilizes the learnable target network as 'target' and weighted moving average to make target network learning smoother and efficiently.

3 Preliminary on SS-LVH

We introduce how to create correlations and training image pairs. Given the target image dataset $\mathcal{X}_D = \{x_i\}_{i=1}^{N}$ and a subset $\mathcal{X}_S = \{x_i\}_{i=1}^{M}$ of $\mathcal{X}_D$, where $x_i \in \mathbb{R}^D$ is the i-th image, C, N, and M are the number of labels, images, and sampled images, respectively.

Label Co-occurrence Correlation Matrix. As shown in Fig. 2, we employ $\mathcal{L}_L = \{L^i\}_{i=1}^{C}$ to calculate the correlation matrix $\mathcal{M}_L \in \mathbb{R}^{C \times C}$ based on the co-occurrence probability of each label, where $\mathcal{L}_L$ is a set of word vectors. For the image x_i, we gather the label probability $p_i \in \mathbb{R}^{C \times 1}$ from the last layer of C-Tran, where p_i denotes the probability of each object contained in the i-th

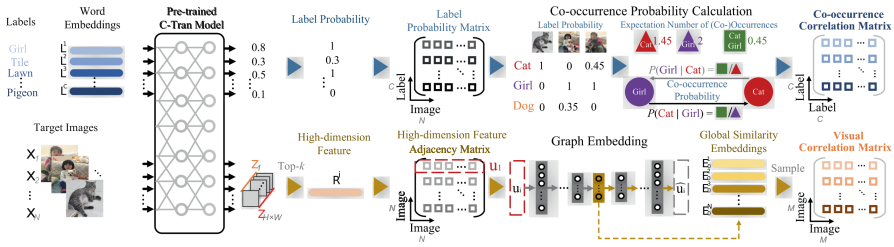


Fig. 2. The generation of the co-occurrence correlation matrix and visual correlation matrix. (Color figure online)

image and it is the i-th column of the label probability matrix $\mathcal{M}_P \in \mathbb{R}^{C\times N}$. Assume that $\mathcal{M}_P(i,j)$ denotes the element of the i-th row and the j-th column in $\mathcal{M}_P$. We change the values of $\mathcal{M}_P(i,j) \geq 0.5$ to 1, reserve the values of $0.5 > \mathcal{M}_P(i,j) \geq 0.3$ (expanding the range of candidates to correct bias), and assign the rest of elements to 0. Note that these settings were determined after we calculated the difference between the generated labels and the actual labels on VOC2007 [6]. To alleviate the sparsity issue caused by a large C, we generalize the method in LAH that regards the occurrence of a label as a discrete state (0 or 1) and calculate the co-occurrence probability $P_{j,i}$, *i.e.*, the probability of the j-th label's occurrence when the i-th label appears, as below.

$$P_{j,i} = \frac{T_{j,i}}{T_i} = \frac{\sum_{k=1}^{N} \mathcal{M}_P(i,k) \times \mathcal{M}_P(j,k)}{\sum_{k=1}^{N} \mathcal{M}_P(i,k)}, \tag{1}$$

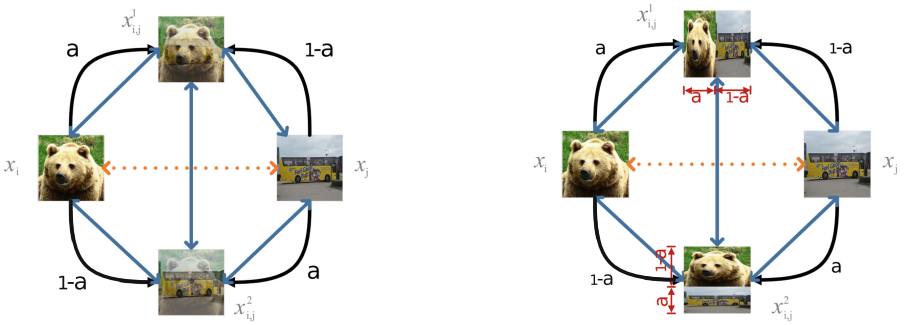
where T_i denotes the expectation number of occurrences for the i-th label and $T_{j,i}$ denotes the expectation number of co-occurrences between the i-th label and the j-th one. Note that although $T_{i,j} = T_{j,i}$, $P_{j,i} \neq P_{i,j}$ when $T_i \neq T_j$. As shown in Fig. 2, only three images contain the *girl* or *cat*, where $P_{cat,girl} \neq P_{girl,cat}$ because $T_{girl} = 2$ (purple triangle) and $T_{cat} = 1.45$ (red triangle). To promote the convergence efficiency and prevent over-fitting, we lower the long-tail effect by using the threshold μ to binarize $P_{j,i}$. Then, we fill $\mathcal{M}_L$ by $P_{j,i}$, which can be described as:

$$\mathcal{M}_L(i,j) = \begin{cases} 0, & \text{if } P_{j,i} \leq \mu, \\ 1, & \text{otherwise .} \end{cases}$$

To further overcome the problem of over-smooth caused by using the correlation matrix in GCNs, we employ the weighted scheme like LAH to determine $\mathcal{M}_L$. The $\mathcal{M}_L$ is described below.

$$\mathcal{M}_L(i,j) = \begin{cases} \frac{\alpha}{\sum_{j=1,i\neq j}^{C} \mathcal{M}_L(i,j)}, & \text{if } i \neq j, \\ 1-\alpha, & \text{otherwise ,} \end{cases} \tag{2}$$

where $\alpha \in (0,1)$. We update a node feature with the effect from α. For instance, a node feature will be more determined by its neighbor nodes when $\alpha \to 1$.



(a) Point-to-point pixel summation and average.

(b) Splicing along the horizontal and vertical.

Fig. 3. Data augmentation strategy. The black lines represent fusion weights. And the blue lines and orange dotted lines represent trusted and untrusted similarities, respectively. (Color figure online)

Image Visual Correlation Matrix. We employ $\mathcal{R}_V = \{R^i\}_{i=1}^M$ to calculate the correlation matrix $\mathcal{M}_V \in \mathbb{R}^{M \times M}$. Nevertheless, we are caught between employing $\mathcal{X}_S$, which results in information loss, and using $\mathcal{X}_D$, which results in a huge cost. Therefore, we learn the embeddings for $\mathcal{X}_D$ and sample the embeddings. As shown in Fig. 2, we collect $\mathcal{X}_D$ through the C-Tran model. Different from conventional features acquired from the convolution layer, our features consist of Z_1 to $Z_{H \times W}$, where each $Z_i \in \mathbb{R}^{1 \times 2048}$ consists of values at the same position for all feature maps, H and W represent the width and height of the feature map, respectively. Since features in C-Tran are generated by the relationship between pixels, we pick the top-k (See Sect. 5) values on each Z_i to construct R^i, *i.e.*, the high-dimension feature of the i-th image. Then, we form the adjacency matrix $\mathcal{M}_A$ by cosine distances between R^i and R^j for the graph embedding learning. Assume that $u_i = \mathcal{M}_A(i, \cdot)$ denotes the vector in the i-th row of $\mathcal{M}_A$. It also represents the similarity between the i-th image and others in $\mathcal{X}_D$. We employ SDNE [21] to encode u_i into the embedding E_0^i, and get the subset $\{E_0^i\}_{i=1}^M$ of $\{E_0^i\}_{i=1}^N$ to calculate $\mathcal{M}_V$, where $\{E_0^i\}_{i=1}^M$ is obtained by random sampling, but preferably in an amount equal to the number of pseudo labels and covering all categories in the target dataset. Each element of $\mathcal{M}_V$ is calculated by cosine distances between E_0^i and E_0^j.

Data Augmentation Strategy. We propose a label similarity transformation strategy (2 patterns) to fuse two images via different weights. As shown in Fig. 3, x_i and x_j are images in $\mathcal{X}_D$, while $x_{i,j}^1$ and $x_{i,j}^2$ are images composed by x_i and x_j with different weights, where weights are a and $1 - a$, and $a \in (0, 1)$ (See Sect. 5). We depict the method of point-to-point pixel summation and average in Fig. 3(a) and the splicing method along with the horizontal and vertical in Fig. 3(b). The label similarity transformation strategy produces more image pairs with the trusted similarity, alleviating the sparsity issue of similar pairs when

C is too large. Note that, since the composite images don't contaminate the correlation matrices, the co-occurrence correlations from the target dataset still are decisive.

4 SS-LVH

SS-LVH learns a nonlinear hash function $f_h : x \mapsto h \in \{-1, 1\}^{\mathcal{K}}$ from input space to Hamming space using CNNs and two GCNs, encoding each image x into a $\mathcal{K}$-bit hash code $h = f_h(x)$. For the target images (untrusted pairs) or composite images (trusted pairs), *i.e.*, x_i and x_j, if their pseudo-multi-labels contain at least one same label, their similarity labels $s_{ij} = 1$. Otherwise, $s_{ij} = 0$. $f_h(x)$ should preserve the similarities, *i.e.*, $\mathcal{S} = \{s_{ij}\}$, in hash codes.

In the representation learning stage, we input $\mathcal{L}_L$ and $\mathcal{M}_L$, pairwise images $\{(x_i, x_j, s_{ij})\}$, and $\mathcal{R}_V$ and $\mathcal{M}_V$ into the label co-occurrence embedding learning branch, the image representation learning branch, and the visual correlation embedding learning branch, respectively. Then, $\{E_1^i\}_{i=1}^C$, E_2, and $\{E_3^i\}_{i=1}^M$ are generated and sent to the dual OGSP component. The fusion results Q_1 and Q_2 are concatenated to the label-visual representation Q. SS-LVH learns Q in the way of BYOL. In the hash learning stage, we fix the learned parameters and learn with the Cauchy distribution loss functions. Finally, SS-LVH transforms Q into a $\mathcal{K}$-dimensional continuous code $\mathcal{Z} \in \mathbb{R}^{\mathcal{K}}$ in the *fc* layer, and then transforms $\mathcal{Z}$ into a $\mathcal{K}$-dimensional hash code by $h = sgn(tanh(\mathcal{Z})) \in \{1, -1\}^{\mathcal{K}}$ in the *fch* layer. Finally, preserving similarity of pairwise images and lowering the quantization error, SS-LVH learns the non-linear hash function $f_h(x)$. The details of each part are described below.

Image Representation Learning. Following LAH, we employ ResNet-101 as the backbone to learn the image representation. For the image x that has been transformed to the dimension of $D = 448 \times 448 \times 3$, *i.e.*, $x \in \mathbb{R}^{448\times448\times3}$, we capture a 2048×14×14-dimensional feature vector from the *conv5_x* layer. Then, we generate $E_2 \in \mathbb{R}^{2048\times1}$ through the *global max-pooling* (GMP) layer.

GCN for Learning of Embeddings. GCN can smooth the features by the given correlation. More specifically, by the propagation of weights, it learns a function f_{gcn} on the graph to achieve feature extraction. For example, on the label co-occurrence embedding learning branch, we assume that $\mathcal{L}_L^{(i)}$ represents the input in the i-th layer and $\mathcal{L}_L^{(i+1)}$ denotes updated node features. The propagation function in each GCN layer is described below.

$$\mathcal{L}_L^{(i+1)} = f_{gcn}(\widehat{\mathcal{M}_L}\mathcal{L}_L^{(i)}\mathcal{W}_L^{(i)}), \tag{3}$$

where $\mathcal{W}_L^{(i)}$ is the weight on the i-th graph layer, $\widehat{\mathcal{M}_L} = \widetilde{D}^{-\frac{1}{2}}\widetilde{\mathcal{M}_L}\widetilde{D}^{-\frac{1}{2}}$ with $\widetilde{\mathcal{M}_L} = \mathcal{M}_L + I_C$ and $\widetilde{D}(i,i) = \sum_j \widetilde{\mathcal{M}_L}(i,j)$. In the implementation, we use two GCN layers with word vectors generated by BERT. The dimensions of the last layer of $\mathcal{L}_L$ and $\mathcal{R}_V$ are designed to match E_2.

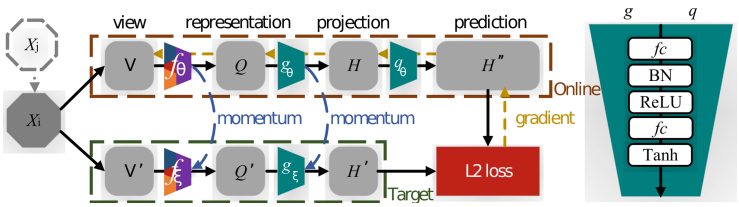


Fig. 4. Self-supervised learning process.

Dual OGSP for Activation. OGSP employs a one-dimensional overlapping window to perform *sum-pooling* over vectors and utilizes Hadmard Product (*i.e.*, $\circ$) to fuse embeddings and image representations. Each fusion result is mapped onto multiple values corresponding to multiple groups (*i.e.*, dotted boxes shown in Fig. 1), resulting in richer information highlighting regions of interest. For the i-th label activated representation Q_L^i, we define that the size of the group is ${}^i\mathcal{G}_L^g$ and the stride is ${}^i\mathcal{G}_L^s$, where $i \in \{1, 2, \ldots, C\}$. Meanwhile, $\mathcal{Q}_L^i = E_1^i \circ E_2$, where $E_1^i, E_2 \in \mathbb{R}^{2048\times 1}$ and $\mathcal{Q}_L^i(k)$ denotes the k-th element of $\mathcal{Q}_L^i$. Thus, the j-th element of Q_L^i is described below.

$$Q_L^i(j) = \sum_{k=1+(j-1)\cdot {}^i\mathcal{G}_L^s}^{{}^i\mathcal{G}_L^g+(j-1)\cdot {}^i\mathcal{G}_L^s} \mathcal{Q}_L^i(k), \tag{4}$$

where $j \in \{1, 2, \ldots, \lceil \frac{2048-{}^i\mathcal{G}_L^g+{}^i\mathcal{G}_L^s}{{}^i\mathcal{G}_L^s} \rceil\}$. Note that when the number of elements is not enough, we fetch elements from the head of vector to fill. Based on this, ${}^{Q_L^i} = [Q_L^i(1); Q_L^i(2); \ldots; Q_L^i(\lceil \frac{2048-{}^i\mathcal{G}_L^g+{}^i\mathcal{G}_L^s}{{}^i\mathcal{G}_L^s} \rceil)]$ and the label semantic similarity representation Q_1 is described below.

$$Q_1 = [Q_L^1; Q_L^2; \ldots; Q_L^C], \tag{5}$$

where $Q_1 \in \mathbb{R}^{\sum_{i=1}^{C} \lceil \frac{2048-{}^i\mathcal{G}_L^g+{}^i\mathcal{G}_L^s}{{}^i\mathcal{G}_L^s} \rceil \times 1}$. In the same way, we define and calculate ${}^i\mathcal{G}_V^g$, ${}^i\mathcal{G}_V^s$, $\mathcal{Q}_V^i = E_3^i \circ E_2$, and Q_V^i. Then, the visual similarity representation Q_2 is described below.

$$Q_2 = [Q_V^1; Q_V^2; \ldots; Q_V^M], \tag{6}$$

where $Q_2 \in \mathbb{R}^{\sum_{i=1}^{M} \lceil \frac{2048-{}^i\mathcal{G}_V^g+{}^i\mathcal{G}_V^s}{{}^i\mathcal{G}_V^s} \rceil \times 1}$. Finally, the label-visual representation Q is described below.

$$Q = [Q_1; Q_2]. \tag{7}$$

Generally, we recommend that $\forall i$, ${}^i\mathcal{G}_L^s = {}^i\mathcal{G}_V^s = \mathcal{G}^s$, ${}^i\mathcal{G}_L^g = {}^i\mathcal{G}_V^g = \mathcal{G}^g$ (See Sect. 5) because $Q \in \mathbb{R}^{\sum_{i=1}^{C+M} \lceil \frac{2048-\mathcal{G}^g+\mathcal{G}^s}{\mathcal{G}^s} \rceil \times 1}$ conduces to the trade-off between two representations.

Self-supervised Learning. As shown in Fig. 4, the framework consists of the *online* (brown dotted frame) and *target* (green dotted frame) networks, whose parameters are θ and ξ respectively. The parameters ξ are an exponential moving

average of θ. Given a target decay rate $\tau \in [0, 1]$, the update after each training step is described below.

$$\xi \leftarrow \tau\xi + (1 - \tau)\theta. \tag{8}$$

The *target* network has the same architecture as the *online* network except for the *prediction* function q. The two *views* V and V' come from image augmentations $\mathcal{T}$ and $\mathcal{T}'$ respectively. In our task, $\mathcal{T}$ and $\mathcal{T}'$ can be conventional methods for a single image, or our similarity transformation strategy for x_i and x_j. f denotes the representation extraction function corresponding to networks in blue, red, orange and purple frames shown in Fig. 1. *Representation* corresponds to Q shown in Fig. 1. $Q = f_\theta(V)$ and $Q' = f_\xi(V')$. g is a *projection* function consists of the BatchNorm layer (BN), $ReLU$, fc layer and $Tanh$, where $Tanh$ is injected to adapt to hash task. $H = g_\theta(Q)$ and $H' = g_\xi(Q')$. q is a *prediction* function and $H'' = q_\theta(H)$, where q has the same architecture as g. Finally, we L_2-normalize $\widehat{H'} = \frac{H'}{\|H'\|_2}$ and $\widehat{H''} = \frac{H''}{\|H''\|_2}$. The loss between the normalized predictions and target projections is described below.

$$\mathscr{L}_{\theta,\xi} = \|\widehat{H'} - \widehat{H''}\|_2^2 = 2 - \frac{2\langle H', H''\rangle}{\|H'\|_2 \cdot \|H''\|_2}. \tag{9}$$

According to Eq. (9), we calculate $\overline{\mathscr{L}_{\theta,\xi}}$ by feeding V to the *target* network and V' to the *online* network. At each training step, the task is to minimize $\widehat{\mathscr{L}_{\theta,\xi}} = \mathscr{L}_{\theta,\xi} + \overline{\mathscr{L}_{\theta,\xi}}$ with respect to θ only. The optimizer of self-supervised learning are described below.

$$\theta \leftarrow Opt(\theta, \nabla_\theta \widehat{\mathscr{L}_{\theta,\xi}}, \eta), \tag{10}$$

where Opt is the stochastic gradient descent optimizer and η is a learning rate. When we use conventional augmentation strategies, we will initialize a low learning rate for label co-occurrence embedding learning branch that enhances activation ability for global visual similarity representations. Contrastively, when we adopt our similarity transformation strategy, we will initialize lower the learning rate for visual correlation embedding learning branch to enhance activation ability for local semantic similarity representations. Finally, we only keep f_θ involving in hash function learning.

Cauchy Loss for Hash Learning. To generate hash codes with high aggregation degree of similar samples within short Hamming distance, we employ Cauchy loss functions used in DCH [1], resulting in the best retrieval performance in Hamming radius ≤ 2.

The Cauchy loss functions consist of the Cauchy cross-entropy loss and the Cauchy quantization loss. For the h_i and h_j corresponding to $\{(x_i, x_j, s_{ij})\}$, the probability function based on the Cauchy distribution is written as:

$$\Gamma(\delta(h_i, h_j)) = \frac{\gamma}{\gamma + \delta(h_i, h_j)}, \tag{11}$$

where $\Gamma(*)$ is well-defined probability function, $\delta(h_i, h_j)$ denotes the Hamming distance between h_i and h_j, γ is the scale hyper-parameter of the Cauchy distribution and controls aggregation degree. Generally, $\gamma = 0.15$.

Assume that $h_i(j)$ is the j-th element of h_i. The sign function $sgn(h_i)$ is described below.

$$sgn(h_i(j)) = \begin{cases} -1, & \text{if } h_i(j) \leq 0, \\ 1, & \text{otherwise .} \end{cases} \tag{12}$$

For the quantization error $\| h - sgn(h) \|$, we combine γ and the Cauchy distribution to describe the prior for each hash code as:

$$\phi_{h_i} = \frac{\gamma}{\gamma + \delta(|h_i|, \mathbb{1})}, \tag{13}$$

where $\mathbb{1} \in \mathbb{R}^{\mathcal{K}}$. To cooperate with continuous relaxation, we set $\delta(h_i, h_j) = \frac{K}{2}(1 - \frac{\langle h_i, h_j \rangle}{\|h_i\|_2 \cdot \|h_j\|_2})$ to approximate the Hamming distance and to optimize the loss function.

Based on Eq. (11) and the logarithm Maximum a Posteriori estimation of the hash codes, the Cauchy cross-entropy loss function $\mathscr{L}_C$ is described below.

$$\mathscr{L}_C = \sum_{s_{ij}} \omega_{ij}(s_{ij} log \frac{\delta(h_i, h_j)}{\gamma} + log(1 + \frac{\gamma}{\delta(h_i, h_j)})), \tag{14}$$

where

$$\omega_{ij} = \begin{cases} |\mathcal{S}|/|\mathcal{S}_s|, & s_{ij} = 1, \\ |\mathcal{S}|/|\mathcal{S}_d|, & s_{ij} = 0, \end{cases}$$

where $\mathcal{S}_s = \{s_{ij} \in \mathcal{S} : s_{ij} = 1\}$ is the set of similar pairs, $\mathcal{S}_d = \{s_{ij} \in \mathcal{S} : s_{ij} = 0\}$ is the set of dissimilar pairs. For $\forall i, j$ and $i \neq j$, if $\exists \mathcal{M}_P(i, k) = \mathcal{M}_P(j, k) = 1$, we obtain $s_{ij} = 1$; otherwise, $s_{ij} = 0$. Meanwhile, according to Eq. (13), the Cauchy quantization loss function $\mathscr{L}_Q$ is described below.

$$\mathscr{L}_Q = \sum_{i=1}^{N} log(1 + \frac{\delta(|h_i|, \mathbb{1})}{\gamma}). \tag{15}$$

According to the deduction of Bayesian learning in DCH [1], the complete hash loss function is described below.

$$\mathscr{L} = \lambda \mathscr{L}_C + (1 - \lambda)\mathscr{L}_Q, \tag{16}$$

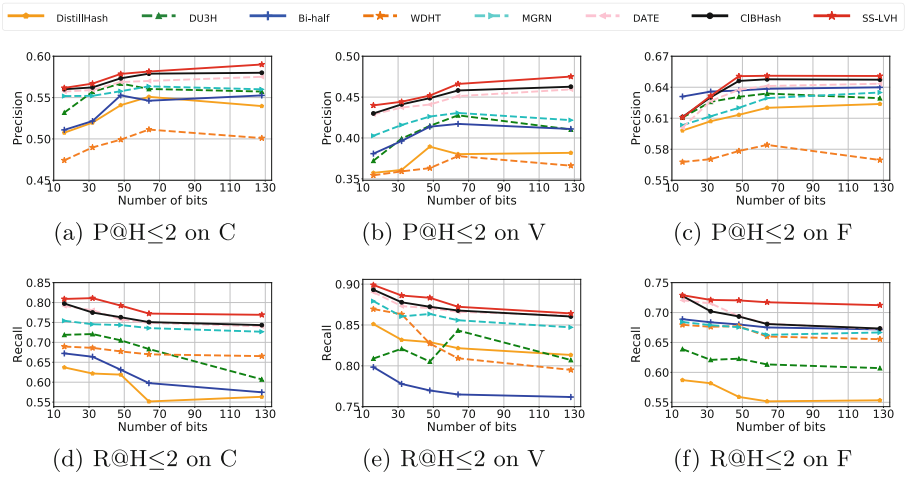
where λ is a hyper-parameter to balance two loss functions.

5 Experiment

Experimental Settings. We select three multi-label image datasets, including MS-COCO [15], VOC2007 [6], and MIRFLICKR-25K [10]. We randomly select 10,000, 4,000, and 5,000 images from three datasets respectively as target datasets to train models. Following parameters in LAH and BYOL, we train all datasets without using hand-crafted labels. Then, we randomly select 5000, 1000, and 1000 images from remaining images as the query set to test models respectively. The classification results in terms of Mean Average Precision

Table 1. MAP of re-ranking within Hamming radius 2 (MAP@H ≤ 2) at different bits on three public multi-label image datasets.

Method	MS-COCO					VOC2007					MIRFLICKR-25K				
	16 bits	32 bits	48 bits	64 bits	128 bits	16 bits	32 bits	48 bits	64 bits	128 bits	16 bits	32 bits	48 bits	64 bits	128 bits
DistillHash [26]	0.605	0.617	0.628	0.630	0.627	0.403	0.410	0.424	0.422	0.420	0.628	0.631	0.633	0.636	0.637
DU3H [27]	0.611	0.620	0.630	0.634	0.633	0.421	0.442	0.448	0.446	0.444	0.636	0.645	0.647	0.646	0.643
TBH [19]	0.607	0.613	0.615	0.618	0.617	0.423	0.441	0.447	0.451	0.448	0.638	0.639	0.642	0.646	0.645
DHNR [23]	0.606	0.609	0.611	0.613	0.611	0.434	0.438	0.439	0.438	0.437	0.624	0.631	0.637	0.647	0.645
Bi-half [14]	0.609	0.616	0.622	0.626	0.626	0.428	0.433	0.438	0.442	0.441	0.640	0.642	0.647	0.650	0.649
WDHT [7]	0.594	0.597	0.608	0.613	0.610	0.389	0.393	0.401	0.411	0.410	0.603	0.616	0.621	0.623	0.616
MGRN [11]	**0.618**	0.621	0.627	0.636	0.636	0.447	0.449	0.452	0.452	0.452	0.631	0.636	0.641	0.645	0.649
DATE [17]	0.611	0.621	0.633	0.639	0.638	0.481	0.488	0.493	0.505	0.507	0.641	0.650	0.656	0.657	0.657
CIBHash [18]	0.617	0.623	0.638	0.641	0.641	0.489	0.504	0.517	0.519	0.518	**0.644**	**0.655**	0.659	0.660	0.659
SS-LVH	0.617	**0.637**	**0.644**	**0.653**	**0.658**	**0.509**	**0.513**	**0.519**	**0.526**	**0.531**	0.639	**0.655**	**0.661**	**0.663**	**0.663**



(a) P@H≤2 on C (b) P@H≤2 on V (c) P@H≤2 on F

(d) R@H≤2 on C (e) R@H≤2 on V (f) R@H≤2 on F

Fig. 5. P@H ≤ 2 and R@H ≤ 2 with different code lengths on the MS-COCO (C), VOC2007 (V) and MIRFLICKR-25K (F) datasets.

(MAP) on MS-COCO, VOC2007, and MIRFLICKR-25K are 0.518, 0.403, and 0.451, respectively when we test datasets by the empirical model. In addition, we employ the methods of the word vectors generation and evaluation metrics used in LAH, where LAH measures the quality of hash codes within Hamming radius 2: MAP within Hamming Radius 2 (MAP@H ≤ 2), Precision curves within Hamming Radius 2 (P@H ≤ 2), and Recall curves within Hamming Radius (R@H ≤ 2).

We compare SS-LVH with nine state-of-the-art label-free hashing methods, including five unsupervised methods (DistillHash [26], DU3H [27], TBH [19], DHNR [23], and Bi-half [14]), two label-embedding-based weakly-supervised methods (WDHT [7] and MGRN [11]), and two SSL methods, *i.e.*, contrastive learning methods (DATE [17] and CIBHash [18]).

Implementation Details. For the label-level co-occurrence embeddings learning, we employ labels of VG-500 and set $C = 500$. For the label-level visual similarity embeddings learning, we set $k = 10$, and $M = 500$ to equal C. For

Table 2. MAP within Hamming radius 2 (MAP@H ≤ 2) of SS-LVH and its variants on three public multi-label image datasets. VB denotes *Visual Correlation Embedding Learning Branch*. GSP denotes *Group Sum-Pooling*. OW denotes *Overlapping Window*. SSL denotes *Self-Supervised Learning*. STS denotes *Label Similarity Transformation Strategy*. ✓ means to enable the component, otherwise disable it.

Order	VB	MFB	GSP	OW	SSL	Tanh	STS	MS-COCO			VOC2007			MIRFLICKR-25K		
								32 bits	64 bits	128 bits	32 bits	64 bits	128 bits	32 bits	64 bits	128 bits
1	✓		✓	✓	✓	✓	✓	0.637	0.653	0.658	0.513	0.526	0.531	0.655	0.663	0.663
2			✓	✓	✓	✓	✓	0.620	0.624	0.623	0.485	0.487	0.481	0.626	0.633	0.618
3	✓	✓			✓	✓	✓	0.441	0.451	0.451	0.354	0.358	0.359	0.406	0.411	0.403
4	✓		✓		✓	✓	✓	0.632	0.648	0.652	0.510	0.519	0.524	0.648	0.650	0.652
5	✓		✓	✓				0.531	0.568	0.570	0.367	0.374	0.377	0.541	0.544	0.546
6	✓		✓	✓	✓			0.623	0.630	0.635	0.483	0.513	0.518	0.634	0.647	0.653
7	✓		✓	✓	✓	✓		0.633	0.639	0.653	0.510	0.522	0.529	0.650	0.660	0.661
8	✓		✓	✓	✓		✓	0.628	0.643	0.649	0.507	0.514	0.527	0.637	0.654	0.659

OGSP, we set $\mathcal{G}^g = 128$ and $\mathcal{G}^s = 32$. For BYOL, we adopt color transformation as the conventional data augmentation strategy. When we input untrusted pairs transformed by the conventional strategy, we initialize $\eta_1 = 0.05$, $\eta_2 = 0.05$, $\eta_3 = 0.1$ and $\eta_4 = 0.03$, where η_1, η_2, η_3 and η_4 denote learning rates of the label co-occurrence embedding learning branch, image representation learning branch, visual correlation embedding learning branch, and other components respectively. When we input trusted pairs using our strategy, we set $a = 0.35$, $\eta_1 = 0.1$, and $\eta_3 = 0.05$. With 1000 epochs, we set the batch sizes to 128 and 32 for the conventional data augmentation strategy and our one respectively, the weight decay to 10^{-6}, and the base target decay rate to $\tau = 0.99$. For the hash learning, we set $\eta_1 = \eta_2 = \eta_3 = 10^{-4}$ and $\eta_5 = 0.05$ with batch size 128, where η_5 is the learning rate of hash learning component. The momentum of optimization is 0.9 and the weight decay is 10^{-4}.

Comparisons with State-of-the-Arts. The MAP@H ≤ 2 of all comparison methods are listed in Table 1, where the underline and bold fonts represent the highest value in the comparison algorithms and all algorithms respectively. These results show that SS-LVH has a stable advantage over other algorithms. Especially at 128 bits, the improvements are 1.7%, 1.3% and 0.4% on MS-COCO, VOC2007 and MIRFLICKR-25K, respectively. Meanwhile, we find that except for contrastive learning methods, the performance of other algorithms will decline when the code length is beyond 32 or 64 bits. We think that the generalization ability derived from visual correlation improves the ability of hash code for carrying semantic information, while our incorporation pattern for the semantic and visual information can enhance this advantage.

To reflect the proportion of retrieved images related to the query image, we show the P@H ≤ 2 performance in Fig. 5(a), Fig. 5(b), and Fig. 5(c). SS-LVH achieves remarkable results on three datasets, and averagely exceeds the runner-up algorithm (*i.e.*, CIBHash) by 0.87%, 0.63% and 0.16% respectively. These results verify the superiority of SS-LVH in the perception of objects and semantic information. In addition, we show the aggregation degree of similar image and the R@H ≤ 2 performance in Fig. 5(d), Fig. 5(e), and Fig. 5(f). SS-LVH is dominant

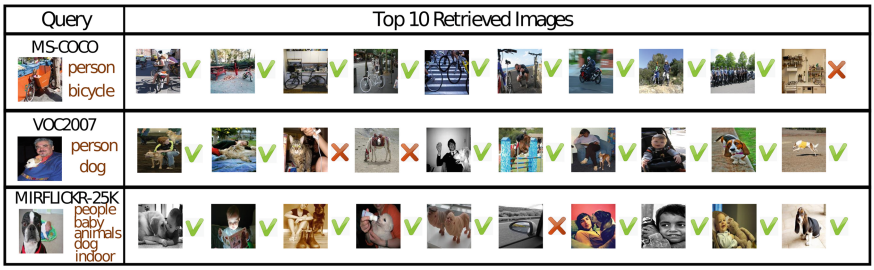
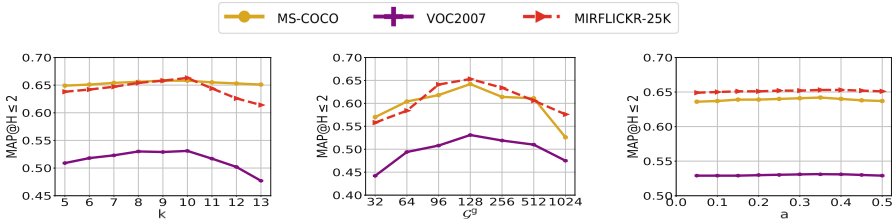


Fig. 6. The top 10 images returned by SS-LVH when we input query images.



(a) varying k (b) varying $\mathcal{G}^g$ (c) varying a

Fig. 7. MAP@H ≤ 2 w.r.t. k, $\mathcal{G}^g$, and a with 128 bits hash codes on the MS-COCO, VOC2007 and MIRFLICKR-25K datasets.

on three datasets and averagely exceeds the runner-up algorithm (*i.e.*, CIBHash) by 0.86%, 1.78% and 3.01% respectively. These results verify the superiority of SS-LVH in the aggregating similar data and the perception of inter-dependency.

To further demonstrate the retrieval effect of SS-LVH, we visualize the top 10 returned images for three query images in Fig. 6.

Ablation Study. To verify contributions of components including the visual correlation embedding learning branch, OGSP, *Tanh*, and label similarity transformation strategy, we list the influence on MAP@H ≤ 2 at different code lengths using different combinations in Table 2. The 1^{st} row denotes the performance of SS-LVH as a benchmark. The performance without the visual correlation embedding learning branch shows in the 2^{nd} row, where the performance averagely decreases by 2.7%, 3.9%, and 3.47% on three datasets, respectively. These declines manifest the necessity of incorporating this visual branch and only employing SSL is not enough to alleviate the noise problem. In addition, the degradation of performance at 128 bits is remarkable. This result confirms that the visual branch helps the hash code carry more semantic information. In the 3^{rd} and 4^{th} rows, we verify the effect of OGSP. Obviously, MFB is not compatible with these SSL components. We believe that this is because the factorized matrices of MFB disturb the spatial information during activation. Furthermore, the overlapping window brings 1.33%, 0.57%, and 1.53% of increments of performance on three datasets. Finally, we notice the influence of SSL components in the 5^{th} to 8^{th} rows. The result in the 5^{th} row means the performance using

pseudo labels without SSL components. Compared with this result, on average, the performance in the 6^{th} row improves **7.3%**, **13.2%**, and **10.1%**, respectively on three datasets. This result shows that the SSL method is important for improvement of hashing performance. Based on SSL, the incorporation of *Tanh* brings 1.23%, 1.57%, and 1.23% of benefits, respectively, and the label similarity transformation strategy also brings 1.07%, 1.13%, and 0.53% of benefits, respectively, on three datasets on average. All in all, our components contribute to performance improvement and the configuration of SS-LVH is optimal.

Hyper-Parameters Sensitivity Analysis. We fix the hyper-parameters that have been verified in other papers and investigate the sensitivity of the designed components' parameters including top-k, $\mathcal{G}^g(\mathcal{G}^s = 32)$, and a. We determine the best hyper-parameter by fixing others with the default values and performing the linear search in candidates. Figure 7 illustrates MAP@H ≤ 2 with 128 bits hash codes on three datasets. According to highest values, SS-LVH can achieve the best retrieval performance when $k = 10$, $\mathcal{G}^g = 128$ and $a = 0.35$.

6 Conclusion

This paper proposes an SS-LVH framework for multi-label image retrieval. Compared with existing methods, we preserve the advantage derived from label co-occurrence correlations and perceive image visual correlation to alleviate the noise problem. The results on three datasets demonstrate the generalization ability and superiority of SS-LVH. Our dual OGSP component, label similarity transformation strategy, and introduction of *Tanh* in BYOL can improve the retrieval performance.

Acknowledgments. This work is supported by the National Natural Science Foundation of China No. 61902135 and No. 62172180, and the Joint Founds of ShanDong Natural Science Funds (Grant No. ZR2019LZH003).

References

1. Cao, Y., Long, M., Liu, B., Wang, J.: Deep cauchy hashing for hamming space retrieval. In: CVPR, pp. 1229–1237 (2018)
2. Cao, Y., Long, M., Wang, J., Liu, S.: Deep visual-semantic quantization for efficient image retrieval. In: CVPR, pp. 916–925. IEEE Computer Society (2017)
3. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.E.: A simple framework for contrastive learning of visual representations. In: ICML, vol. 119, pp. 1597–1607. PMLR (2020)
4. Chen, Z., Wei, X., Wang, P., Guo, Y.: Multi-label image recognition with graph convolutional networks. In: CVPR, pp. 5177–5186 (2019)
5. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) NAACL-HLT, pp. 4171–4186. Association for Computational Linguistics (2019)

6. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. Int. J. Comput. Vis. **88**(2), 303–338 (2010)
7. Gattupalli, V., Zhuo, Y., Li, B.: Weakly supervised deep image hashing through tag embeddings. In: CVPR, pp. 10375–10384 (2019)
8. Grill, J., et al.: Bootstrap your own latent - a new approach to self-supervised learning (2020)
9. Huang, C., Yang, S., Pan, Y., Lai, H.: Object-location-aware hashing for multi-label image retrieval via automatic mask learning. IEEE Trans. Image Process. **27**(9), 4490–4502 (2018)
10. Huiskes, M.J., Lew, M.S.: The MIR flickr retrieval evaluation. In: SIGMM, pp. 39–43 (2008)
11. Jin, L., Li, Z., Pan, Y., Tang, J.: Weakly-supervised image hashing through masked visual-semantic graph-based reasoning. In: MM, pp. 916–924 (2020)
12. Lai, H., Yan, P., Shu, X., Wei, Y., Yan, S.: Instance-aware hashing for multi-label image retrieval. IEEE Trans. Image Process. **25**(6), 2469–2479 (2016)
13. Lanchantin, J., Wang, T., Ordonez, V., Qi, Y.: General multi-label image classification with transformers, pp. 16478–16488 (2021)
14. Li, Y., van Gemert, J.: Deep unsupervised image hashing by maximizing bit entropy. In: EAAI, pp. 2002–2010 (2021)
15. Lin, T.-S., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
16. Liu, Y., et al.: Deep self-taught hashing for image retrieval. IEEE Trans. Cybern. **49**(6), 2229–2241 (2019)
17. Luo, X., et al.: A statistical approach to mining semantic similarity for deep unsupervised hashing. In: MM, pp. 4306–4314. ACM (2021)
18. Qiu, Z., Su, Q., Ou, Z., Yu, J., Chen, C.: Unsupervised hashing with contrastive information bottleneck. In: IJCAI, pp. 959–965. ijcai.org (2021)
19. Shen, Y., et al.: Auto-encoding twin-bottleneck hashing. In: CVPR, pp. 2815–2824 (2020)
20. Song, J., He, T., Gao, L., Xu, X., Shen, H.T.: Deep region hashing for efficient large-scale instance search from images. CoRR abs/1701.07901 (2017)
21. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: SIGKDD, pp. 1225–1234 (2016)
22. Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., Xu, W.: CNN-RNN: a unified framework for multi-label image classification. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016, pp. 2285–2294. IEEE Computer Society (2016)
23. Wang, Y., Song, J., Zhou, K., Liu, Y.: Unsupervised deep hashing with node representation for image retrieval. Pattern Recogn. **112**, 107785 (2021)
24. Wu, Z., Shen, C., van den Hengel, A.: Wider or deeper: revisiting the resnet model for visual recognition. Pattern Recogn. **90**, 119–133 (2019)
25. Xie, Y., Liu, Y., Wang, Y., Gao, L., Wang, P., Zhou, K.: Label-attended hashing for multi-label image retrieval, pp. 955–962 (2020)
26. Yang, E., Liu, T., Deng, C., Liu, W., Tao, D.: DistillHash: unsupervised deep hashing by distilling data pairs. In: CVPR, pp. 2946–2955 (2019)
27. Zhang, W., Wu, D., Zhou, Y., Li, B., Wang, W., Meng, D.: Deep unsupervised hybrid-similarity hadamard hashing. In: MM, pp. 3274–3282 (2020)