Label-Attended Hashing for Multi-Label Image Retrieval

Yanzhao Xie[1] , **Yu Liu**[1*] , **Yangtao Wang**[1] , **Lianli Gao**[2] , **Peng Wang**[1] and **Ke Zhou**[1]
[1]Huazhong University of Science and Technology
[2]The University of Electronic Science and Technology of China
{yzxie, liu_yu, ytwbruce, wp_hust, k.zhou}@hust.edu.cn, lianli.gao@uestc.edu.cn

Abstract

For the multi-label image retrieval, the existing hashing algorithms neglect the dependency between objects and thus fail to capture the attention information in the feature extraction, which affects the precision of hash codes. To address this problem, we explore the inter-dependency between objects through their co-occurrence correlation from the label set and adopt Multi-modal Factorized Bilinear (MFB) pooling component so that the image representation learning can capture this attention information. We propose a Label-Attended Hashing (LAH) algorithm which enables an end-to-end hash model with inter-dependency feature extraction. LAH first combines Convolutional Neural Network (CNN) and Graph Convolution Network (GCN) to separately generate the image representations and label co-occurrence embeddings, then adopts MFB to fuse these two modal vectors, finally learns the hash function with a Cauchy distribution based loss function via backpropagation. Extensive experiments on public multi-label datasets demonstrate that (1) LAH can achieve the state-of-the-art retrieval results and (2) the usage of co-occurrence relationship and MFB not only promotes the precision of hash codes but also accelerates the hash learning. GitHub address: https://github.com/IDSM-AI/LAH

1 Introduction

Similarity hash code has been widely used in large-scale image retrieval owing to its lightweight storage (compact binary code) and efficient comparison (exclusive-OR) [Liu *et al.*, 2019; 2020]. For classic image hashing, correctly recognizing the object from an image is an important factor to promote the retrieval precision. However, it becomes more challenging for multi-label image retrieval where each image contains more objects. Several studies [Lai *et al.*, 2016; Song *et al.*, 2017; Huang *et al.*, 2018] specifically designed for multi-label image retrieval recognize each object in isolation and then fuse these features to learn a hash model, which is essentially limited by ignoring the complex topology structure between objects and thus fails to further improve the precision of retrieval. For example, a basketball will appear in a court with a higher probability than a football. The existing hashing methods have not benefited from this prior information.

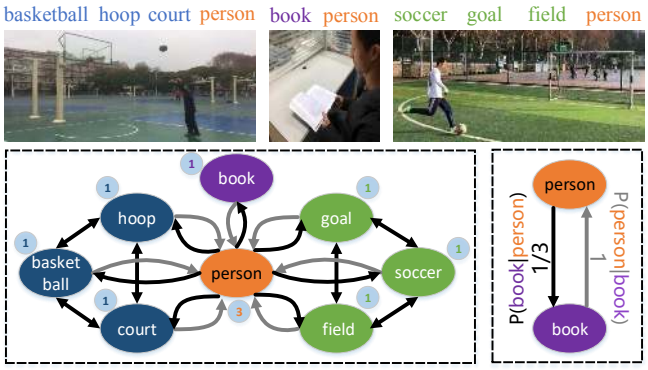


Figure 1: We first collect the times of objects that appear in the dataset according to the labels, then determine the co-occurrence probabilities based on the co-occurrence times of the appeared objects. We model inter-dependency by a directed graph in the lower left frame and express co-occurrence probabilities on edges in the lower right frame. Note that the mutual conditional probabilities between two objects are asymmetrical. As shown in the lower right frame, *person* appears three times, *book* appears only once and they co-occur once, so the co-occurrence conditional probabilities between these two objects are respectively $1/3$ and 1.

To address this problem, there are two key challenges: (1) how to construct an ideal topology structure and what kind of dependency should be expressed, (2) how to use the topology information to learn image representations in the hash task via an end-to-end manner.

Based on this, we first explore the correlation between objects according to their co-occurrence probability [Wang *et al.*, 2017] which is collected from labels via attention mechanism. As shown in Figure 1, we leverage the directed graph structure to capture and explore the label dependency between objects. The reason why we model a directed graph is that the mutual conditional probabilities between two objects (labels) are asymmetrical. Take the three images in Figure 1 as example, when *person* appears in the image, *book* will oc-

*Corresponding Author

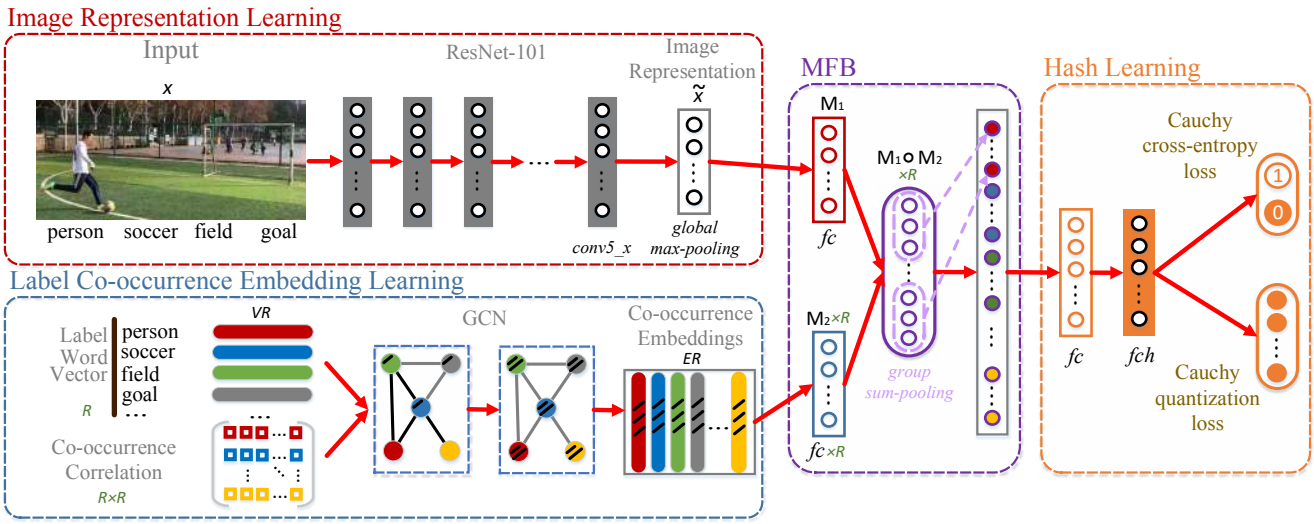


Figure 2: The architecture of the proposed Label-Attended Hashing (LAH), which is comprised of four key components: (1) a CNN, *i.e.*, ResNet-101, for learning deep representation of each image, (2) a GCN for learning label co-occurrence embedding, where R denotes the number of labels, *VR* denotes the set of label vectors and *ER* denotes the set of co-occurrence embeddings, (3) a MFB for efficiently fusing image representation and label co-occurrence embeddings, where M_1 denotes a transformation of an image representation and M_2 denotes the transformation of a co-occurrence embedding (there are R different M_2 corresponding to R labels) and (4) cross-entropy loss and quantization loss based on the Cauchy distribution which respectively aim at similarity hash learning and hash codes quality controlling.

cur with probability 33.33%. However, in the condition of *book* appearing, *person* will certainly occur with probability 100%.

For the second challenge, we first combine Graph Convolutional Network (GCN) [Zhou *et al.*, 2018] and Word2Vector method [Pennington *et al.*, 2014] to learn the label co-occurrence embeddings, where these embeddings can well express the features of co-occurrence correlation. Although these features come from the label word vectors, they function as the classifiers in the fusion of image features and help learn the image features that embody the co-occurrence correlation. ML-GCN [Chen *et al.*, 2019] adopts a dot product (DP) in this fusion process, which achieves good results in terms of image classification. For the feature extraction of hash tasks that require more effective and more attention information, we use our improved MFB instead of DP. On the one hand, the MFB component improves fusion efficiency by adding fully connected (*fc*) layers with more flexible fitting space. On the other hand, it generates more fusion parameters by group sum-pooling to both promote the fusion precision and extend the attention representations.

Based on the above work, we propose a label-attended hash (LAH) model which can capture multi-label image features and fast achieve hash mapping according to the label dependency in the end-to-end pipeline. For the label co-occurrence embedding, we take both the word vectors and co-occurrence correlation as input, then use GCN to calculate the label embeddings which implicitly represent the co-occurrence relationship. Meanwhile, we use ResNet-101 [He *et al.*, 2016] to generate an image representation by means of global max-pooling. In the following, we fuse co-occurrence embeddings and the image representation via MFB to learn an end-to-end hash model. As for the hash function, we introduce the state-of-the-art hash function consisting of a quantization loss and a pairwise cross-entropy loss based on the Cauchy distribution to obtain the optimal result, both of which conduce to concentrate relevant images to be within a small Hamming ball. The overall deep architecture is shown in Figure 2. The proposed LAH model can be trained end-to-end along red line by backpropagation. Extensive experiments on public multi-label datasets demonstrate LAH can generate highly concentrated and compact hash codes and achieve the state-of-the-art retrieval results. LAH also has higher representation learning efficiency compared with ML-GCN.

2 Label-Attended Hashing

In the implementation, we assume that there are N samples $\{x_i|i = 1, 2, \cdots, N\}$ in the training set, where $L(x_i)$ denotes the label set of the i-th sample and s_{ij} denotes the similarity between x_i and x_j (s_{ij}=1 if x_i is similar to x_j while s_{ij}=0 if they are dissimilar). Specifically, in the multi-label scenario, we stipulate $s_{ij} = 1$ if $L(x_i) \cap L(x_j) \neq \emptyset$, otherwise $s_{ij} = 0$. In addition, we assume that there are R objects $\{r_g|g = 1, 2, \cdots, R\}$ in the whole label set and $ER = \{E(r_g)|g = 1, 2, \cdots, R\}$ where $E(r_g)$ denotes the co-occurrence embedding of the g-th object. The architecture for LAH is shown in Figure 2, where the pipeline along the red arrow is the backbone of our algorithm which can be trained by backpropagation. When training our model in this backbone, we input pairs of $\{(x_i, x_j, s_{ij})\}$ and *ER*, where $D(x)$ and $D(E(r)) \times R$ respectively denote the dimension of x and *ER*. After the image representation learning stage, x is transformed into a vector $\tilde{x}$ with $D(\tilde{x})$ dimension which will be sent to MFB along with $E(r_g)$ to generate a $1 \times R$ vector Z after completing R times fusion. In the last hash learning stage, LAH transforms Z into K-dimensional continu-

ous code $\tilde{Z} \in \mathbb{R}^K$ for each image x, and then transforms $\tilde{Z}$ into K-dimensional hash code by $h = sgn(tanh(\tilde{Z})) \in \{-1, 1\}^K$ in the *fc* hash (*fch*) layer. Finally, with Cauchy cross-entropy loss and Cauchy quantization loss, LAH learns a non-linear hash function $\mathcal{F}(x) : x \to h \in \{-1, 1\}^K$, which encodes each sample x into compact K-bit hash code. The details of each component are described below.

2.1 Image Representation Learning

Any CNN based models can complete the feature extraction of an image in our architecture. In our experiments, following ML-GCN, we choose ResNet-101 as the model. Based on this, for an image x with the resolution $D(x) = 448 \times 448$, we can obtain a $2048 \times 14 \times 14$-dimensional feature vector from the "$conv5\_x$" layer. At last, we adopt global max-pooling $\mathcal{F}_{gmp}$ to generate the image-level feature $\tilde{x}$:

$$\tilde{x} = \mathcal{F}_{gmp}(\mathcal{F}_{cnn}(x; \theta))$$

where θ denotes the CNN parameters and $D(\tilde{x}) = 2048$.

2.2 GCN for Label Co-occurrence Embedding Learning

The motivation of introducing GCN is that it can learn the relationship representation from the feature descriptions according to the input of the relationship. Essentially, it learns a propagation function $\mathcal{F}_{gcn}$ on the graph via weight propagation to complete feature extraction. We use the GloVe [Pennington *et al.*, 2014] model to convert the object (word description in the label set) into vector $V(r)$, where $V^c \in \mathbb{R}^{R \times D(V(r))}$ represents the input in c-th layer and $D(V(r))$ represents the dimension of $V(r)$. The input of relationship is the correlation matrix $A \in \mathbb{R}^{R \times R}$, and the updated node features are denoted as $V^{c+1} \in \mathbb{R}^{R \times D(V(r))'}$. Each GCN layer propagation function is described as:

$$V^{c+1} = \mathcal{F}_{gcn}(\hat{A}V^cW^c)$$

where $\hat{A} = \tilde{D}^{-\frac{1}{2}}(A + I_R)\tilde{D}^{-\frac{1}{2}}$ with $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ and $\tilde{A} = A + I_R$.

In our experiments, we use two GCN layers, *i.e.*, $E(r) = V^{c+2}$. Referring to ML-GCN, we adopt the data-driven method to construct matrix A which is the key of co-occurrence learning. Specifically, in order to determine each element of matrix A, we have to first collect the occurrence times as well as co-occurrence times of each object according to the label set. Let T_i, T_j respectively denote the occurrence times of r_i, r_j in the label set and T_{ij} (which equals T_{ji}) denote the co-occurrence times of these two objects. Then, we model the label correlation dependency in the form of conditional probability, *i.e.*,

$$P_{ij} = P(r_i|r_j) = \frac{T_{ij}}{T_j}$$

which denotes the probability of occurrence of r_i when r_j appears. Note that, as shown in Figure 1, $T_i \neq T_j$, $P_{ij} \neq P_{ji}$. Based on this, $A_{ij} = P_{ji}$ where A_{ij} denotes the i-th row and j-th column element of A.

However, in the implementation, we find if directly using this simple correlation, the overall data distribution may exhibit a long-tail effect owing to the rare co-occurrence objects which become noise and affect the model convergence. To promote the efficiency and prevent over-fitting, we use the threshold τ to binarize the matrix A, the operation can be written as:

$$\bar{A}_{ij} = \begin{cases} 0, & \text{if } P_{ji} \leq \tau \\ 1, & \text{otherwise} \end{cases}$$

where $\bar{A}$ is the binary correlation matrix. In addition, there exactly exists the over-smoothing problem [Chen *et al.*, 2019] when using the above binary correlation matrix to train the GCN model, which may result in that the generated hash codes become indistinguishable. Therefore, similar to ML-GCN, we adopt the weighted scheme and draw the final correlation matrix $\mathcal{A}$ as:

$$\mathcal{A}_{ij} = \begin{cases} \frac{q}{\sum_{j=1 \cap i \neq j}^{R} \bar{A}_{ij}}, & \text{if } i \neq j \\ 1 - q, & \text{otherwise} \end{cases}$$

where $q \in (0, 1)$. In this way, we can update a node feature by choosing a fixed weight q. For example, a node feature will rely more on itself if using a smaller q. Otherwise, its feature will be determined by other neighbor nodes.

2.3 MFB for Fusion

The MFB is one of the keys to achieve high precision with fast learning efficiency for LAH. The motivation of improving this component to build the LAH fusion mechanism lies in that both the multi-label image representation learning and the label co-occurrence embedding learning implicitly adopt the attention mechanism, while MFB can achieve the highest efficiency and best performance when conducting multi-modal co-attention fusion. Specifically, on the one hand, we utilize Hadmard product and group sum-pooling instead of the DP to increase the interaction of vector elements among different modalities, which promotes the precision. On the other hand, by means of sum-pooling, it reduces over-fitting and parameter explosion caused by the increasing interaction and thus accelerates the model convergence.

Given two feature vectors in different modalities, *i.e.*, image representation $\tilde{x} \in \mathbb{R}^{D(\tilde{x})}$ and co-occurrence embedding $E(r) \in \mathbb{R}^{D(E(r))}$, we also draw lessons from tricks for uni-modal data [Li *et al.*, 2017] to reduce high computational cost and a risk of over-fitting. For the feature of the i-th object, the multi-modal bilinear model with two low-rank matrices is defined as follows:

$$\begin{aligned} z_i &= \tilde{x}^T U_i V_i^T E(r) = \sum_{d=1}^{k} \tilde{x}^T u_d v_d^T E(r) \\ &= \mathbb{1}^T (U_i^T \tilde{x} \circ V_i^T E(r)) \end{aligned} \tag{1}$$

where k is the latent dimensionality of the factorized matrices $U = [u_1, \cdots, u_k] \in \mathbb{R}^{D(\tilde{x}) \times k}$ and $V = [v_1, \cdots, v_k] \in \mathbb{R}^{D(E(r)) \times k}$, $\circ$ is the Hadmard product, *i.e.*, the element-wise multiplication of two vectors, $\mathbb{1} \in \mathbb{R}^k$ is an all-one vector.

In practice, as shown in Figure 2, we adopt two parallel k-dimensional *fc* layers (red and blue *fc* layers shown in MFB framework) to complete this transform respectively, where $M_1 = U_i^T \tilde{x}$ and $M_2 = V_i^T E(r)$.

According to the total number of object categories, we need to get R vector z as input to match the subsequent hash mapping. Another purpose of this design is to provide better explanation for the representation of each object. Therefore, to obtain the output $Z = [z_1, \cdots, z_R]$, the weights to be learned are two three-order tensors $\tilde{U} = [U_1, \cdots, U_R] \in \mathbb{R}^{D(\tilde{x}) \times k \times R}$ and $\tilde{V} = [V_1, \cdots, V_R] \in \mathbb{R}^{D(E(r)) \times k \times R}$. Without loss of generality, we can reformulate $\tilde{U}$ and $\tilde{V}$ as 2-D matrices $\tilde{U} \in \mathbb{R}^{D(\tilde{x}) \times (k \times R)}$ and $\tilde{V} \in \mathbb{R}^{D(E(r)) \times (k \times R)}$ respectively. At last, we obtain the result of group sum-pooling as below:

$$Z = \mathcal{F}^G_{sum-pooling}(\tilde{U}^T \tilde{x} \circ \tilde{V}^T E(r), k)$$

where $\frac{k}{G} \in [1, k]^{Z+}$ and the function $\mathcal{F}^G_{sum-pooling}(*, k)$ means using a one-dimensional non-overlapped window with the size $\frac{k}{G}$ to perform sum-pooling over $*$. Different from [Yu *et al.*, 2017], we improve this work where all the R identical vectors in $\tilde{U}$ are generated from $\tilde{x}$ while the vectors in $\tilde{V}$ correspond to the elements of *ER* transformed by *fc* layer.

2.4 Cauchy Distribution for Hash Learning

Due to the diversity of retrieval target for the multi-label image, it becomes a tricky task to generate hash codes with small distance between similar data and large distance between dissimilar data in the Hamming space. Luckily, Cauchy distribution can adjust the location of data peak to embody the focus of attention. According to the verification of DCH [Cao *et al.*, 2018], the hash function based on Cauchy distribution can solve the problem that the conventional $sigmoid$ function brings low aggregation degree of similar samples within short Hamming distance, and it can achieve better results when Hamming radius ≤ 2 in terms of Mean Average Precision (MAP), Precision and Recall. Therefore, we adopt Cauchy distribution based function as our LAH hash function to promote the aggregation degree of similar data within small Hamming radius by jointly preserving similarity of pairwise images and controlling the quantization error.

Assume that $h_i, h_j \in \{-1, 1\}^K$ respectively denote the output of x_i and x_j after the *fch* layer while $\{(x_i, x_j, s_{ij}) : s_{ij} \in \mathcal{S}\}$ denotes the input. Based on the Cauchy distribution, we design the probability function:

$$\sigma(\delta(h_i, h_j)) = \frac{\gamma}{\gamma + \delta(h_i, h_j)} \tag{2}$$

where $\sigma(*)$ is well-defined probability function, $\delta(h_i, h_j)$ denotes the Hamming distance between h_i and h_j, γ is the scale parameter of the Cauchy distribution. Note that the smaller γ, the higher aggregation degree within short Hamming radius. We repeatedly conduct extensive experiments and choose $\gamma = 0.15$.

To learn high-quality hash codes and control the quantization error $\| h_i - sgn(h_i) \|$ resulting from continuous relaxation, we combine parameter γ (Equation (2)) and Cauchy distribution to design prior for each hash code as follows:

$$\mathcal{P}_{h_i} = \frac{\gamma}{\gamma + \delta(|h_i|, \mathbb{1})} \tag{3}$$

where $\mathbb{1} \in \mathbb{R}^K$. In order to cooperate with the continuous relaxation based on Cauchy distribution, we adopt $\delta(h_i, h_j) = \frac{K}{2}(1 - cos(h_i, h_j))$ to normalize Euclidean distance as the approximation of Hamming distance to ease the optimization. The normalized Euclidean distance compares each pair of continuous codes on a unit sphere, while the Hamming distance compares each pair of hash codes on a unit hyper-cube.

According to the deduction of Bayesian learning in DCH and Equation (2), the Cauchy cross-entropy loss $\mathcal{L}_{cce}$ is derived as:

$$\mathcal{L}_{cce} = \sum_{s_{ij}} \omega_{ij} (s_{ij} log \frac{\delta(h_i, h_j)}{\gamma} + log(1 + \frac{\gamma}{\delta(h_i, h_j)})) \tag{4}$$

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 and $\mathcal{S}_d = \{s_{ij} \in \mathcal{S} : s_{ij} = 0\}$ is the set of dissimilar pairs.

In addition, according to equation (3), the Cauchy quantization loss $\mathcal{L}_{cq}$ is

$$\mathcal{L}_{cq} = \sum_{i=1}^{N} log(1 + \frac{\delta(|h_i, \mathbb{1}|)}{\gamma}) \tag{5}$$

Based on above equations, the completed hash function

$$\mathcal{L} = \lambda \mathcal{L}_{cce} + (1 - \lambda) \mathcal{L}_{cq}$$

where λ is a hyper-parameter to trade-off two loss functions. Note that our sign function is designed as

$$sgn(h_i) = \begin{cases} 1, & h_i > 0, \\ -1, & \text{otherwise.} \end{cases}$$

where h_i is a element in vector $h \in \mathbb{R}^K$. This binarization will not severely affect the retrieval result because it have been adjusted by the quantization loss.

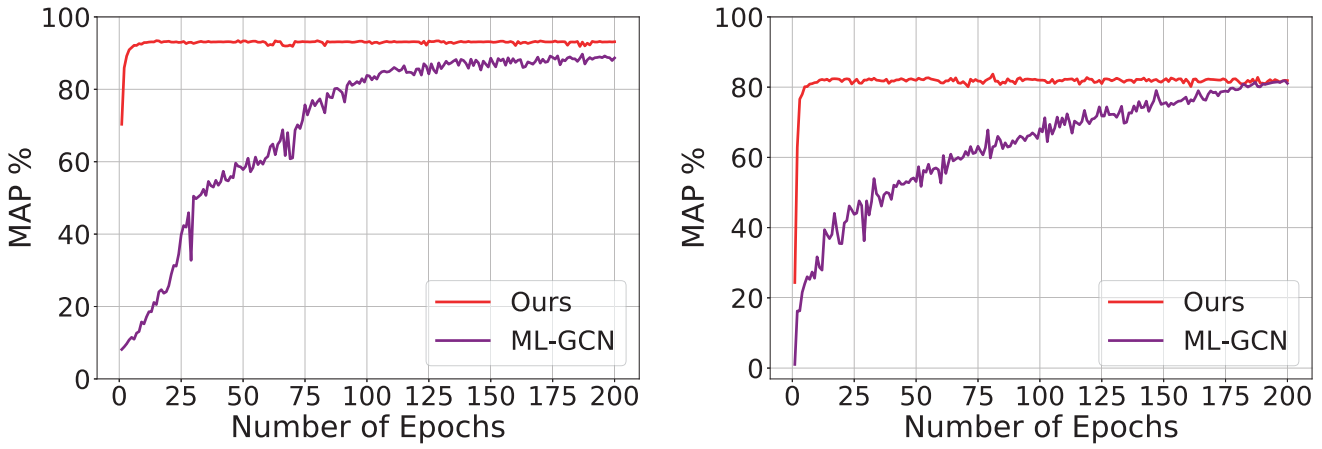
3 Experiments

3.1 Datasets

VOC2007. [Everingham *et al.*, 2010] consists of 9,963 multi-label images and 20 object classes. On average, each image is annotated with 1.5 labels. Note that we transform 0 to 1 in label dataset.

MS-COCO. [Lin *et al.*, 2014] is a popular multiple object dataset for image recognition, segmentation and captioning, which contains 118,287 training images, 40,504 validation images and 40,775 test images, where each image is labeled by some of the 80 semantic concepts.

FLICKR25K. [Huiskes and Lew, 2008] is a collection of 25,000 multi-label images belonging to 24 unique provided labels, and each image is annotated by 4.7 labels on average. We randomly select 2,000 images as the test set. The remaining images are used as the retrieval images, where we randomly select 10,000 images as the training set.



(a) VOC2007 (b) MS-COCO

Figure 3: MAP on test set with epoch increasing on training set.

Method	VOC2007					MS-COCO					FLICKR25K				
	16 bits	32 bits	48 bits	64bits	128bits	16 bits	32 bits	48 bits	64bits	128bits	16 bits	32 bits	48 bits	64bits	128bits
DSRH	0.5741	0.6194	0.6608	0.6907	0.6956	0.6185	0.6707	0.6992	0.7266	0.7184	0.6691	0.6849	0.7044	0.7339	0.7330
IAH	0.5997	0.6618	0.6884	0.7830	0.8727	0.7374	0.7676	0.7903	0.8393	0.8414	0.7994	0.8317	0.8366	0.8361	0.8381
OLAH	0.6109	0.6699	0.7174	0.7890	0.8852	0.7830	0.8229	0.8512	0.8848	0.8935	**0.8016**	0.8662	0.8890	0.8905	0.8918
RCDH	**0.6177**	0.6739	0.7206	0.7979	0.8863	0.7839	0.8381	0.8616	0.8911	0.8940	0.7983	0.8491	0.8812	0.8909	0.8924
T-MLZSH	0.5812	0.6473	0.7099	0.7710	0.7697	0.6074	0.6376	0.6779	0.6902	0.6730	0.6484	0.6695	0.6805	0.6816	0.6806
DCH	0.5779	0.6556	0.7576	0.7326	0.7033	0.8010	0.8576	0.8521	0.8299	0.7836	0.7837	0.8681	0.8694	0.8407	0.7911
GCNH	0.6001	0.6637	0.7375	0.7996	0.8559	0.7889	0.8494	0.8662	0.8777	0.8811	0.7621	0.8493	0.8727	0.8766	0.8813
DistillHash	0.5716	0.6583	0.7459	0.7861	0.7780	0.6279	0.6432	0.6612	0.6837	0.6756	0.6964	0.7056	0.7259	0.7241	0.7226
LAH	0.5984	**0.6745**	**0.7797**	**0.8318**	**0.9191**	**0.8101**	**0.8628**	**0.8730**	**0.9006**	**0.9019**	0.7834	**0.8782**	**0.9043**	**0.9107**	**0.9260**

Table 1: Mean Average Precision within Hamming Radius 2 (MAP@H≤2) at Different Bits on Three Benchmark Datasets.

3.2 Evaluation Metrics

Following classic settings [Cao *et al.*, 2018], we report the three standard evaluation metrics to measure the quality of hash codes within Hamming radius 2: Mean Average Precision within Hamming Radius 2 (MAP@H≤2), Precision curves within Hamming Radius 2 (P@H≤2), and Recall curves within Hamming Radius 2 (R@H≤2).

3.3 Implementation Details

For label co-occurrence embedding learning, our LAH consists of two GCN layers with output dimensionality of 1024 and 2048 ($D(E(r)) = 2048$), where the initial label vector ($D(V(r)) = 300$) is generated by GloVe trained on the Wikipedia dataset. For the label expressed by multiple words, we adopt the average of embeddings for all words to present the initial label vector. As for the correlation matrix, we set $\tau = 0.4$ and $q = 0.2$. In the part of image representation learning, we adopt ResNet-101 pre-trained on ImageNet [Deng *et al.*, 2009] to function as the backbone before MFB component, where mini-batch size is fixed as 256 and the raw images (input) are random resized into 448×448 using random horizontal flips. In the part of MFB, without otherwise stated, we set $k = 350$ for all datasets. For fair comparisons with other algorithms, we set $G = 350$.

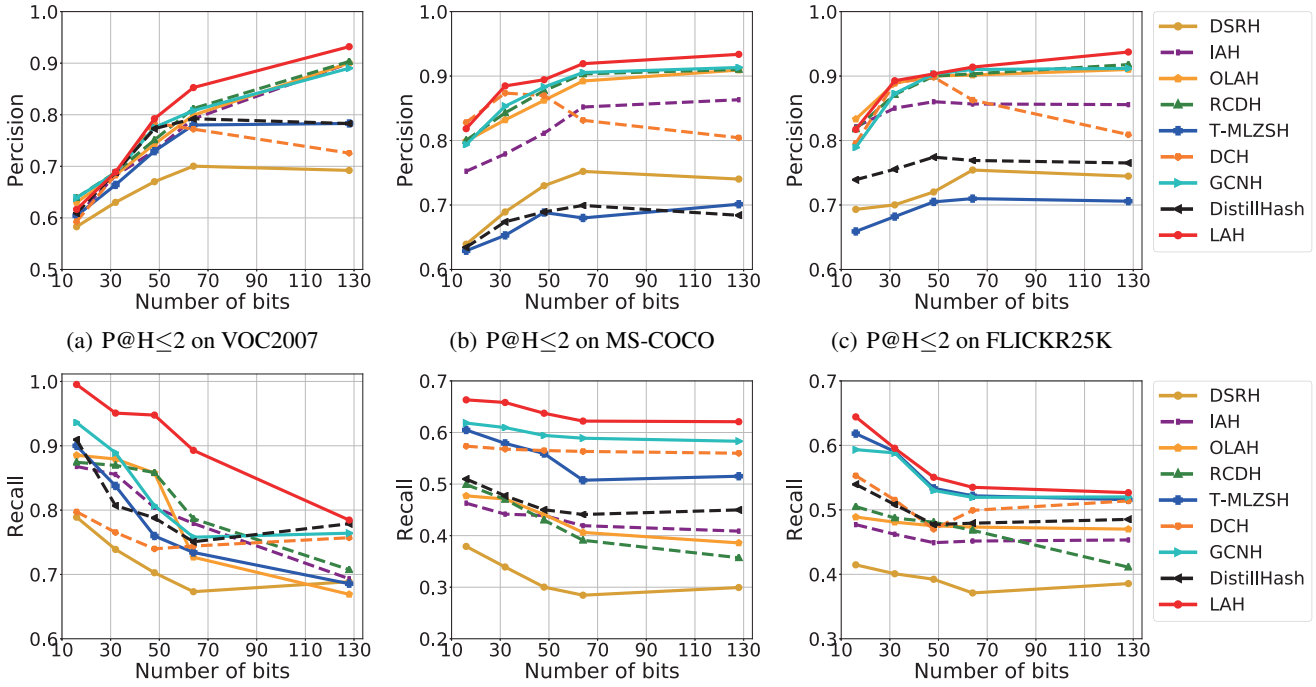
In the part of *fch*, we set the model parameters ($\gamma = 1$ and $\lambda = 0.55$) of LAH by cross-validation. We fine-tune ResNet-101 as well as train GCN, MFB and *fch* all through backpropagation. The processing is implemented using PyTorch[1]. For network optimization, Stochastic Gradient Descent (SGD) [Amari, 1993] is used as the optimizer with 0.9 momentum and 10^{-4} weight decay. Note that all the results are obtained within 20 epochs.

3.4 Results

Learning Efficiency

To show the efficiency of MFB, we compare the classification precision of LAH (using the improved feature extraction part based on ML-GCN) with that of ML-GCN in each epoch. For fair comparisons, we use the same training parameters, loss function and datasets (*i.e.*, VOC2007 and MS-COCO) as ML-GCN. As shown in Figure 3, LAH has already converged on the 15-th and 17-th epoch, and obtained the best MAP of 93.43% and 82.40% respectively. However, at the same epoch, ML-GCN never converges and its MAP scores are respectively 72.91% and 38.39% lower than LAH on VOC2007 and MS-COCO.

[1]https://pytorch.org/



(a) P@H≤2 on VOC2007 (b) P@H≤2 on MS-COCO (c) P@H≤2 on FLICKR25K

(d) R@H≤2 on VOC2007 (e) R@H≤2 on MS-COCO (f) R@H≤2 on FLICKR25K

Figure 4: P@H≤2 and R@H≤2 with different code lengths on VOC2007, MS-COCO and FLICKR25K.

Comparisons with State-of-the-Arts

We compare the retrieval performance of LAH with existing multi-label hashing methods including DSRH [Zhao *et al.*, 2015], IAH [Lai *et al.*, 2016], OLAH [Huang *et al.*, 2018], RCDH [Ma *et al.*, 2018], T-MLZSH [Zou *et al.*, 2019], and the state-of-the-art hashing methods including DCH [Cao *et al.*, 2018], GCNH [Zhou *et al.*, 2018] and DistillHash [Yang *et al.*, 2019].

The Mean Average Precision within Hamming Radius 2 (MAP@H≤2) of all comparison methods are listed in Table 1. Experimental results show LAH has stable advantages over other algorithms at different code lengths and averagely outperforms the runner-up by 1.37%, 0.77% and 1.21% in terms of MAP@H≤2 on VOC2007, MS-COCO and FLICKR25K respectively.

Note that, we find both the MAP@H≤2 values of DCH and DistillHash will greatly decline at 64 bits and 128 bits, but this phenomenon becomes not obvious on those hash algorithms specially designed for multi-label datasets. This shows that the well-designed multi-label hash algorithms can carry more semantic information with longer hash codes. Fortunately, our proposed LAH owns this ability. At the same time, it can be seen that GCNH, whose difference from others is to use GCN to complete the image feature extraction, also owns this ability. We believe GCN has played a vital role in this process and our LAH also benefits from it. In addition, LAH outperforms GCNH owing that we introduce the co-occurrence relationship into GCN.

The performance of Precision within Hamming Radius 2 (P@H≤2) reflects the proportion of retrieved images related to the query image. As shown in Figure 4(a), (b) and (c), LAH achieves the highest P@H≤2 results on all the three benchmark datasets at different code lengths, and averagely exceeds the runner-up by 1.12%, 0.94% and 0.22% on VOC2007, MS-COCO and FLICKR25K respectively. This result verifies the superiority of LAH in feature extraction, because preciser classification features can bring preciser hash codes.

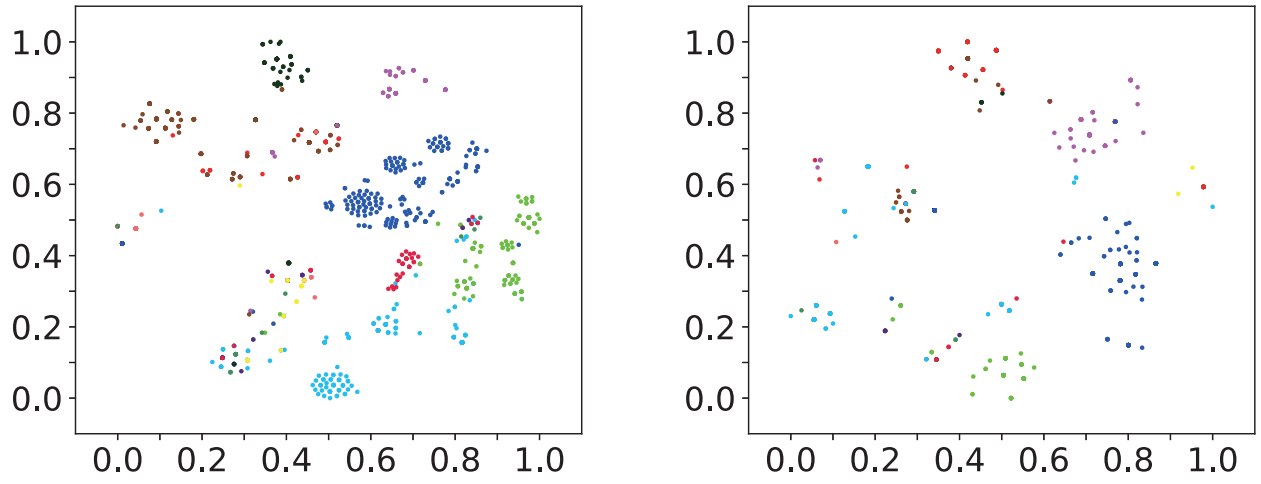
Note that R@H≤2 reflects the aggregation degree of similar images in the Hamming space. As shown in Figure 4(d), (e) and (f), LAH also achieves the highest R@H≤2 results on all the three datasets, and averagely exceeds the runner-up by 6.45%, 4.16% and 1.36% respectively. This result verifies the superiority of LAH in aggregating similar data, which benefits from the design of hash function. The visualization results are displayed in Section 3.6.

3.5 Ablation Study

We present the influence on the performance of LAH using different components. Word embedding method (WEM) is a LAH component used to acquire initial word vectors for labels. We compare the performance under GloVe [Pennington *et al.*, 2014] with other popular counterparts including GoogleNews [Mikolov *et al.*, 2013] and FastText [Joulin *et al.*, 2016]. Fusion method (FM) is a LAH component used to fuse the label information and image feature. We compare the performance under MFB and DP which is used in ML-GCN. As shown in Table 2, the precision results under MFB at different code lengths are respectively 12.03%, 2.40% and 6.32% higher than that under DP on VOC2007, MS-COCO and FLICKR25K. In addition, LAH produces better perfor-

WEM	FM	VOC2007					MS-COCO					FLICKR25K				
		16 bits	32 bits	48 bits	64bits	128bits	16 bits	32 bits	48 bits	64bits	128bits	16 bits	32 bits	48 bits	64bits	128bits
GloVe	MFB	0.5984	**0.6745**	**0.7797**	**0.8318**	**0.9191**	**0.8101**	**0.8628**	0.8730	**0.9006**	**0.9019**	0.7834	**0.8782**	**0.9043**	**0.9107**	**0.9260**
	DP	0.6098	0.6713	0.7476	0.8010	0.9015	0.7984	0.8118	0.8502	0.8841	0.9001	0.7553	0.7804	0.8889	0.8971	0.9017
GoogleNew	MFB	0.5906	0.6339	0.6917	0.8073	0.8553	0.8007	0.8386	0.8620	0.8990	0.9009	0.7470	0.7366	0.8593	0.8783	0.9002
	DP	0.5014	0.6341	0.6125	0.6523	0.7986	0.8026	0.8511	0.8371	0.8985	0.9007	0.7024	0.7887	0.8307	0.8632	0.8982
FastText	MFB	**0.6133**	0.6726	0.6986	0.7547	0.8265	0.8010	0.8381	0.8645	0.8852	0.9007	0.7486	0.7264	0.8513	0.8661	0.8983
	DP	0.6018	0.6249	0.6704	0.7053	0.8141	0.7998	0.8376	**0.8891**	0.8666	0.8913	0.7374	0.7490	0.8066	0.8223	0.8768
Neither WEM nor FM		0.5779	0.6556	0.7576	0.7326	0.7033	0.8010	0.8576	0.8521	0.8299	0.7836	**0.7837**	0.8681	0.8694	0.8407	0.7911

Table 2: Mean Average Precision within Hamming Radius 2 (MAP@H$\leq$2) of LAH and Its Variants on Three Benchmark Datasets. WEM is *Word Embedding Method*. FM is *Fusion Method*. MFB is *Multi-modal Factorized Bilinear Pooling*. DP is *Dot Product*.



(a) LAH (b) RCDH

Figure 5: The t-SNE visualization of hash codes on VOC2007.

mance using GloVe than other WEMs, and respectively outperforms them by 3.69% and 4.06% over GoogleNew and FastText. In addition, LAH averagely outperforms the approach that uses neither WEM nor FM by 5.67% in all cases. From these results, it can been seen that the combination of components we designed is effective and optimal.

3.6 Visualization Study

Figure 5 shows the t-SNE visualization of the hash codes learnt by LAH and our baseline RCDH on VOC2007 dataset. LAH can not only better distinguish different categories but also gather the same categories more compactly. Although RCDH is equally excellent, its effect is not as good as LAH.

4 Conclusion

In this paper, we propose LAH, a label-attended hashing algorithm for multi-label image retrieval. LAH adopts co-occurrence probability over the label set to explore the inter-dependency between objects and uses GCN to learn the co-occurrence embeddings. In the following, LAH utilizes MFB to fuse the image representations and label co-occurrence embeddings in an end-to-end manner, which integrates the multi-modal co-attention as well as promotes the learning efficiency. Extensive experiments on VOC2007, MS-COCO and FLICKR25K demonstrate LAH has a high convergence rate and can generate high-quality hash codes and achieve better retrieval results than the state-of-the-art hashing methods.

Acknowledgments

This work is supported by the National Natural Science Foundation of China No.61902135 and the Innovation Group Project of the National Natural Science Foundation of China No.61821003. Wuhan is a great city whose residents are all heroes. We will surely overcome the COVID-19 and harvest freedom in the end.

References

[Amari, 1993] Shun-ichi Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(3):185–196, 1993.

[Cao *et al.*, 2018] Yue Cao, Mingsheng Long, Bin Liu, and Jianmin Wang. Deep cauchy hashing for hamming space retrieval. In *CVPR*, pages 1229–1237, 2018.

[Chen *et al.*, 2019] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *CVPR*, pages 5177–5186, 2019.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[Everingham *et al.*, 2010] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, jun 2010.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[Huang *et al.*, 2018] Chang-Qin Huang, Shang-Ming Yang, Yan Pan, and Hanjiang Lai. Object-location-aware hashing for multi-label image retrieval via automatic mask learning. *IEEE Trans. Cybernetics*, 27(9):4490–4502, 2018.

[Huiskes and Lew, 2008] Mark J. Huiskes and Michael S. Lew. The MIR flickr retrieval evaluation. In *SIGMM*, pages 39–43, 2008.

[Joulin *et al.*, 2016] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hervé Jégou, and Tomas Mikolov. Fasttext.zip: Compressing text classification models. *CoRR*, abs/1612.03651, 2016.

[Lai *et al.*, 2016] Hanjiang Lai, Pan Yan, Xiangbo Shu, Yunchao Wei, and Shuicheng Yan. Instance-aware hashing for multi-label image retrieval. *IEEE Trans. Image Processing*, 25(6):2469–2479, 2016.

[Li *et al.*, 2017] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Factorized bilinear models for image recognition. In *ICCV*, pages 2098–2106, 2017.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, pages 740–755, 2014.

[Liu *et al.*, 2019] Yu Liu, Jingkuan Song, Ke Zhou, Lingyu Yan, Li Liu, Fuhao Zou, and Ling Shao. Deep self-taught hashing for image retrieval. *IEEE Trans. Cybernetics*, 49(6):2229–2241, 2019.

[Liu *et al.*, 2020] Yu Liu, Yangtao Wang, Ke Zhou, Yujuan Yang, and Yifei Liu. Semantic-aware data quality assessment for image big data. *Future Gener. Comput. Syst.*, 102:53–65, 2020.

[Ma *et al.*, 2018] Cheng Ma, Zhixiang Chen, Jiwen Lu, and Jie Zhou. Rank-consistency multi-label deep hashing. In *ICME*, pages 1–6, 2018.

[Mikolov *et al.*, 2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[Song *et al.*, 2017] Jingkuan Song, Tao He, Lianli Gao, Xing Xu, and Heng Tao Shen. Deep region hashing for efficient large-scale instance search from images. *CoRR*, abs/1701.07901, 2017.

[Wang *et al.*, 2017] Zhouxia Wang, Tianshui Chen, Guanbin Li, Ruijia Xu, and Liang Lin. Multi-label image recognition by recurrently discovering attentional regions. In *ICCV*, pages 464–472, 2017.

[Yang *et al.*, 2019] Erkun Yang, Tongliang Liu, Cheng Deng, Wei Liu, and Dacheng Tao. Distillhash: Unsupervised deep hashing by distilling data pairs. In *CVPR*, pages 2946–2955, 2019.

[Yu *et al.*, 2017] Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In *ICCV*, pages 1839–1848, 2017.

[Zhao *et al.*, 2015] Fang Zhao, Yongzhen Huang, Liang Wang, and Tieniu Tan. Deep semantic ranking based hashing for multi-label image retrieval. In *CVPR*, pages 1556–1564, 2015.

[Zhou *et al.*, 2018] Xiang Zhou, Fumin Shen, Li Liu, Wei Liu, Liqiang Nie, Yang Yang, and Heng Tao Shen. Graph convolutional network hashing. *IEEE transactions on cybernetics*, 2018.

[Zou *et al.*, 2019] Qin Zou, Zheng Zhang, Ling Cao, Long Chen, and Song Wang. Transductive zero-shot hashing for multi-label image retrieval. *CoRR*, abs/1911.07192, 2019.