Deep Joint Semantic-Embedding Hashing

Ning Li[1], Chao Li[1], Cheng Deng[1]*, Xianglong Liu[2], Xinbo Gao[1]
[1] School of Electronic Engineering, Xidian University, Xi'an 710071, China
[2] Beihang University, Beijing 100191, China
ningli2017@gmail.com, li_chao@stu.xidian.edu.cn, {chdeng, xbgao}@mail.xidian.edu.cn,
xlliu@nlsde.buaa.edu.cn

Abstract

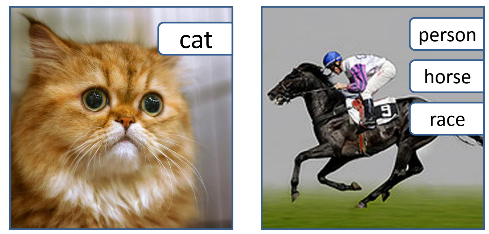
Hashing has been widely deployed to large-scale image retrieval due to its low storage cost and fast query speed. Almost all deep hashing methods do not sufficiently discover semantic correlation from label information, which results in the learned hash codes less discriminative. In this paper, we propose a novel Deep Joint Semantic-Embedding Hashing (**DSEH**) approach that consists of *LabNet* and *ImgNet*. Specifically, *LabNet* is explored to capture abundant semantic correlation between sample pairs and supervise *ImgNet* from both semantic level and hash codes level, which is conductive to the generated hash codes being more discriminative and similarity-preserving. Extensive experiments on three benchmark datasets show that the proposed model outperforms current state-of-the-art methods.

1 Introduction

Due to the explosive increase of high-dimensional media data in search engines and social networks, approximate nearest neighbor (ANN) search for large-scale datasets has attracted more and more attention. Among existing ANN techniques, hashing has become the most popular and effective one due to its fast query speed and low memory cost [Deng *et al.*, 2015a; 2015b], which aims to map high-dimensional data into compact binary codes and preserve their original similarities.

Recently, deep hashing methods [Xia *et al.*, 2014; Lai *et al.*, 2015; Cao *et al.*, 2017; Yang *et al.*, 2017; 2018; Li *et al.*, 2018] have gained state-of-the-art performance due to their powerful ability of feature learning by using deep network architecture, with which we can build more accurate similarity relationship and then generate more discriminative hash codes. Compared with unsupervised deep hashing methods, supervised ones can achieve better performance with the aid of label information. Even so, how to sufficiently discover the semantic correlation from label information is still a crucial issue to be addressed. In this paper, we mainly focus on extracting abundant semantic correlation from label information with deep neural network.

*Corresponding author



(a) ImageNet (b) NUS-WIDE

Figure 1: Single-label dataset vs. multi-label dataset.

Actually, existing supervised hashing methods do not rationally exploit label information of samples, almost all of which only simply construct the similarity affinity matrix of sample pairs [Xia *et al.*, 2014; Li *et al.*, 2015; Liu *et al.*, 2016a]. As shown in Fig. 1a, for the ImageNet dataset, each sample is annotated by single label, where the similarity relationship between samples is very sparse, *i.e.*, the number of similar pairs is much smaller than the number of dissimilar pairs, which will result in that the learned hash codes cannot preserve the original similarity relationship effectively. To tackle this problem, HashNet [Cao *et al.*, 2017] alleviates such data imbalance by adjusting the weights of similar pairs. However, the optimal weights cannot be easily obtained, which limits its feasibility to real-world retrieval system. For NUS-WIDE dataset, as shown in Fig. 1b, each sample is annotated with multiple labels, which can provide high level semantic information and complex similarity relationship. Unfortunately, multiple labels in current methods are oversimplified to single-label case, which removes many useful semantic information and cannot maintain the original similarity relationship of sample pairs. Therefore, either single-label or multi-label dataset, we should capture more abundant semantic correlation to indicate the accurate similarity relationship between samples and produce more discriminative hash codes.

In this paper, we propose a novel Deep Joint Semantic-Embedding Hashing method, namely DSEH, in which both *LabNet* and *ImgNet* are end-to-end networks containing semantic layers and hash layers. In *LabNet*, label information are projected into common semantic space and common Hamming space for exploring abundant semantic features and discriminative hash codes, respectively. In *ImgNet*, an image is embedded into the common semantic space and common

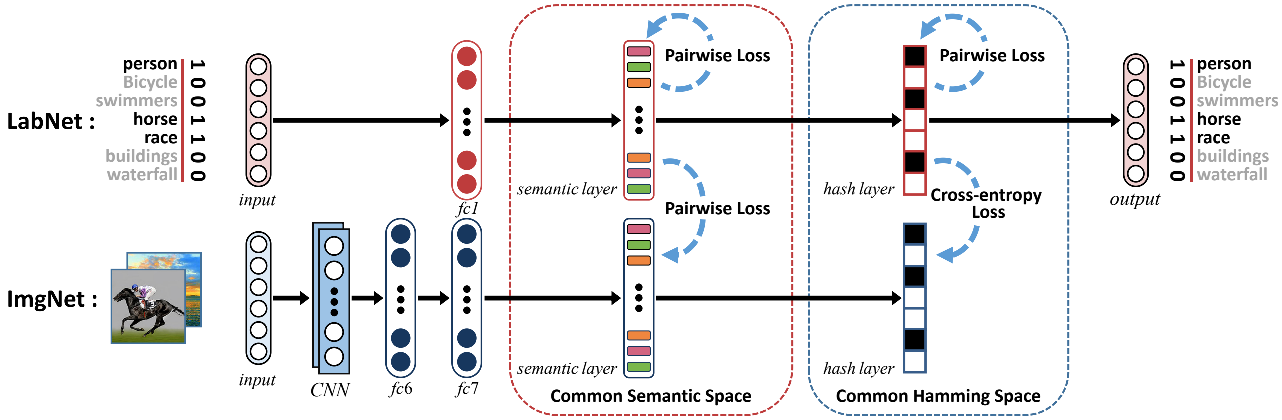


Figure 2: The framework of our proposed DSEH.

Hamming space. By exploiting the learned semantic correlation and hash codes in *LabNet* as supervised information and transferring them to *ImgNet* with the form of two constraints, more accurate semantic correlation can be discovered and thus discriminative hash codes can be generated. Extensive experiments, conducted on three popular datasets including single-label and multi-label ones, demonstrate the proposed DSEH outperforms state-of-the-art hashing approaches.

The main contributions of our DSEH are summarized as follows. 1) We exploit a novel architecture for deep hashing, consisting of *LabNet* and *ImgNet*, where common semantic space and common Hamming space are built across the networks. 2) We utilize a couple of constrains to build a relationship between *LabNet* and *ImgNet* from semantic feature level and hash code level. 3) We adopt an alternative training strategy to jointly optimize the parameters of these two networks, and produce the optimal hash codes.

2 Related Work

Existing hashing methods can be roughly categorized into unsupervised [Gionis *et al.*, 1999; Weiss *et al.*, 2009; Gong *et al.*, 2013; Liu *et al.*, 2016b] and supervised hashing [Liu *et al.*, 2012; Shen *et al.*, 2015; Deng *et al.*, 2014; 2016; Liu *et al.*, 2016a; 2016a; Deng *et al.*, 2018]. Unsupervised hashing methods learn hash functions from unlabeled data. Locality Sensitive Hashing (LSH) [Gionis *et al.*, 1999] uses random projections as hash function. Graph-based hashing [Liu *et al.*, 2011] learns appropriate hash codes by discovering inherent neighborhood structure. Supervised hashing methods incorporate semantic label or relevance information to improve the quality of hash codes. Binary Reconstruction Embedding (BRE) [Kulis and Darrell, 2009] designs hash functions by minimizing the squared errors between the original distances and the reconstructed distances in Hamming space. Supervised Hashing with Kernels (KSH) [Liu *et al.*, 2012] learns to build compact binary codes by minimizing the Hamming distances on similar pairs and maximizing those on dissimilar pairs.

Deep hashing methods have been presented recently, which achieve promising performance due to the powerful arbitrary nonlinear representation of deep neural network. With the help of this structure, CNNH [Xia *et al.*, 2014] learns approximate hash codes from the pairwise similarity regularization first, then tries to learn feature representation and hash function based on the hash codes in the first stage. DNNH [Lai *et al.*, 2015] and DPSH [Li *et al.*, 2015] integrate feature learning and hashing learning into a unified end-to-end network to improve the discrimination of hash codes. DSH [Liu *et al.*, 2016a] groups training data into similar pairs and dissimilar pairs to generate similarity correlation and controls the quantization error. One further study, HashNet [Cao *et al.*, 2017] uncovers the inherent problem caused by data imbalance of some single-label dataset and alleviates this drawback by adjusting the weights of semantic correlation matrix. However, the data imbalance remains a challenge and almost all of these methods do not or little exploits semantic information to generate semantic correlation from label information directly.

3 Proposed DSEH

Fig. 2 shows the flowchart of the proposed method, which mainly consists of two parts: *LabNet* and *ImgNet*. *LabNet* is an end-to-end fully connected deep neural network, where a semantic layer and a hash layer are built to generate semantic features and hash codes from label information. Meanwhile, *ImgNet* consists of a convolution neural network with a semantic layer and a hash layer, which is used to learn hash codes of the input images.

3.1 Problem Formulation

In similarity retrieval scenario, given a dataset $\mathcal{O} = \{\boldsymbol{o}_i\}_{i=1}^n$, $\boldsymbol{o}_i = (\boldsymbol{v}_i, \boldsymbol{l}_i)$, where $\boldsymbol{v_i} \in \mathbb{R}^{1\times d_v}$ is a feature vector of the ith sample, which could be hand-crafted feature, deep feature, or raw pixels of an image. $\boldsymbol{l}_i = [l_{i1}, \cdots, l_{ic}]$ is the label annotations assigned to $\boldsymbol{o}_i$, where c is the number of classes. $\boldsymbol{o}_i$ and $\boldsymbol{o}_j$ are associated with similarity label s_{ij}, where $s_{ij} = 1$ implies $\boldsymbol{o}_i$ and $\boldsymbol{o}_j$ are similar, or otherwise $s_{ij} = 0$. In our setting, we define $s_{ij} = 1$ if $\boldsymbol{o}_i$ and $\boldsymbol{o}_j$ share at least one label, and $s_{ij} = 0$ if $\boldsymbol{o}_i$ and $\boldsymbol{o}_j$ have no common label. The goal of deep hashing is to learn nonlinear hash function, *i.e.*, $f : \boldsymbol{o} \mapsto \boldsymbol{h} \in \{-1, 1\}^K$, to encode each sample $\boldsymbol{o}$ into compact K-bit hash code $\boldsymbol{h}$, such that the original similarity between sample pairs can be well preserved.

For two binary hash codes $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$, their Hamming distance $dis_H(\boldsymbol{h}_i, \boldsymbol{h}_j)$ and inner product $\langle \boldsymbol{h}_i, \boldsymbol{h}_j\rangle$ can be formulated as:

$$dis_H(\boldsymbol{h}_i, \boldsymbol{h}_j) = \frac{1}{2}(K - \langle \boldsymbol{h}_i, \boldsymbol{h}_j \rangle). \tag{1}$$

If the inner product of two binary codes is small, their Hamming distance should be large, and vice versa. Given the hash codes $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$, the similarity probability between $\boldsymbol{o}_i$ and $\boldsymbol{o}_j$ is defined as a likelihood function:

$$p\left(s_{ij}|\boldsymbol{h}_i, \boldsymbol{h}_j\right) = \begin{cases} \sigma\left(\boldsymbol{h}_i^\top \boldsymbol{h}_j\right), & s_{ij} = 1 \\ 1 - \sigma\left(\boldsymbol{h}_i^\top \boldsymbol{h}_j\right), & s_{ij} = 0 \end{cases} \tag{2}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function. Similar to logistic regression, we can see that the smaller hamming distance $dist_H(\boldsymbol{h}_i, \boldsymbol{h}_j)$ is, the larger their inner product $\langle \boldsymbol{h}_i, \boldsymbol{h}_j \rangle$ is. A smaller condition probability $P(1|\boldsymbol{h}_i, \boldsymbol{h}_j)$ implies $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$ should be similar; otherwise, a larger condition probability $P(0|\boldsymbol{h}_i, \boldsymbol{h}_j)$ means $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$ should be dissimilar. Thus, quantifying the similarity relationship between hash codes in Hamming space can be transformed into calculating the inner product of original hash codes.

Similar to hash learning, replacing two features $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ in Eq. (2), the similarity between two features can also be calculated. The larger $\langle \boldsymbol{f}_i, \boldsymbol{f}_j \rangle$ is, the greater the similarity of them is, and vice versa. The similarity probability of $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ can be expressed as likelihood function:

$$p\left(s_{ij}|\boldsymbol{f}_i, \boldsymbol{f}_j\right) = \begin{cases} \sigma\left(\boldsymbol{f}_i^\top \boldsymbol{f}_j\right), & s_{ij} = 1 \\ 1 - \sigma\left(\boldsymbol{f}_i^\top \boldsymbol{f}_j\right), & s_{ij} = 0 \end{cases} \tag{3}$$

3.2 *LabNet* Learning

For discovering the abundant semantic correlation from label information, our *LabNet* is constrained in both semantic space and Hamming space. Pairwise correlation loss in these two spaces should be concerned. Let $f(\boldsymbol{l}_i; \theta^l)$ denote embedding labels for point i, and θ^l is the parameter of *LabNet*.

Different from generating supervised information only in the Hamming space in most exiting methods, a new semantic space is constructed in our method, with which similarity relationship can be well preserved at semantic level. For all the instances in semantic space, given features $\boldsymbol{F}^l = \{\boldsymbol{f}_i^l\}_{i=1}^n$ and pairwise similarity labels $\mathcal{S} = \{s_{ij}\}$, the logarithm Maximum a Posterior (MAP) estimation of semantic features $\boldsymbol{F}^l = \left[\boldsymbol{f}_1^l, \cdots, \boldsymbol{f}_N^l\right]$ can be expressed as:

$$\begin{aligned} \log p(F^l|\mathcal{S}) &\propto \log p(\mathcal{S}|F^l)p(F^l) \\ &= \sum_{s_{ij}\in\mathcal{S}} \log p(s_{ij}|\boldsymbol{f}_i^l, \boldsymbol{f}_j^l)p(\boldsymbol{f}_i^l, \boldsymbol{f}_j^l) \end{aligned} \tag{4}$$

where $p(\mathcal{S}|F^l)$ is the likelihood function, and $p(F^l)$ is the prior distribution. By taking the negative log-likelihood of the observed pairwise labels in $\mathcal{S}$, we can frame the following optimization problem as:

$$\begin{aligned} \min_{F^l,\theta^l} \mathcal{J}_1 &= -\log p\left(\mathcal{S}|\, F^l\right) \\ &= -\sum_{s_{ij}\in\mathcal{S}} \left(s_{ij}\boldsymbol{f}_i^{l\top}\boldsymbol{f}_j^l - \log(1+\exp(\boldsymbol{f}_i^{l\top}\boldsymbol{f}_j^l))\right) \end{aligned} \tag{5}$$

It is easy to find that the above optimization problem can make semantic features $\boldsymbol{F}^l$ to preserve the original similarity relationship in semantic space.

Then, semantic features are embedded into Hamming space to produce compact binary codes which also need to keep the original similarities. The MAP estimation of hash codes $\boldsymbol{H}^l = \left[\boldsymbol{h}_1^l, \cdots, \boldsymbol{h}_N^l\right]$ can be represented as:

$$\begin{aligned} \log p(H^l|\mathcal{S}) &\propto \log p(\mathcal{S}|H^l)p(H^l) \\ &= \sum_{s_{ij}\in\mathcal{S}} \log p(s_{ij}|\boldsymbol{h}_i^l, \boldsymbol{h}_j^l)p(\boldsymbol{h}_i^l, \boldsymbol{h}_j^l). \end{aligned} \tag{6}$$

When substituting Eq. (2) into MAP estimation in Eq. (6), the problem can be formulated as:

$$\begin{aligned} \min_{H^l,\theta^l} \mathcal{J}_2 &= -\log p\left(\mathcal{S}|\, H^l\right) \\ &= -\sum_{s_{ij}\in\mathcal{S}} \left(s_{ij}\boldsymbol{h}_i^{l\top}\boldsymbol{h}_j^l - \log(1+\exp(\boldsymbol{h}_i^{l\top}\boldsymbol{h}_j^l))\right) \end{aligned} \tag{7}$$

Furthermore, in order to promote the hash value discretization, binary regularization should be considered additionally, which can be formulated as follow:

$$\min_{H^l,\theta^l} \mathcal{J}_3 = \sum_{s_{ij}\in\mathcal{S}} \left(\||\boldsymbol{h}_i^l| - \mathbf{1}\|_1 + \||\boldsymbol{h}_j^l| - \mathbf{1}\|_1\right) \tag{8}$$

where $\mathbf{1} \in \mathbb{R}^K$ is the vector of ones, and $\|\cdot\|_1$ denotes the ℓ_1-norm of a vector.

Finally, to maintain the semantic information during the training of *LabNet*, the achieved hash codes from Hamming space is mapped to original label. Therefore, the output of *LabNet* can be written as:

$$\hat{\boldsymbol{Y}}^l = \boldsymbol{W}^\top \boldsymbol{H}^l + \boldsymbol{b} \tag{9}$$

where $\hat{\boldsymbol{Y}}^l$ is the predicted label of output, and $\boldsymbol{W}$ is the mapping weight. To minimize the distance between the predict label $\hat{\boldsymbol{y}}_i^l$ and ground truth $\boldsymbol{y}_i^l$, the least squares loss is adopted as follows:

$$\min_{\hat{\boldsymbol{Y}}^l,\theta^l} \mathcal{J}_4 = \sum_{i=1}^N \|\boldsymbol{y}_i^l - \hat{\boldsymbol{y}}_i^l\|_2^2 = \sum_{i=1}^N \|\boldsymbol{y}_i^l - \boldsymbol{w}^\top\boldsymbol{h}_i^l - \boldsymbol{b}\|_2^2 \tag{10}$$

where $\|\cdot\|_2$ is l_2 norm of a vector.

The overall objective function for *LabNet* can be written as follows:

$$\min_{F^l,H^l,\theta^l} \mathcal{L}_{\mathrm{Lab}} = \mathcal{J}_1 + \alpha\mathcal{J}_2 + \beta\mathcal{J}_3 + \gamma\mathcal{J}_4 \tag{11}$$

where α, β, γ are the hyper-parameters corresponding to the loss function, respectively.

3.3 *ImgNet* Learning

ImgNet is supervised by *LabNet* from semantic features as well as hash codes. Let $g(\boldsymbol{v}_i; \theta^v)$ be the learned image feature for the ith samples, where θ^v is the network parameter of *ImgNet*.

In the common semantic space between *LabNet* and *ImgNet*, if the sample pairs $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are similar, their corresponding features $\boldsymbol{f}_i^v$ and $\boldsymbol{f}_j^v$ should also be similar. Supervised by the semantic feature of *LabNet*, the semantic feature F^v of *ImgNet* can be depicted as:

$$\begin{aligned}\min_{F^v,\theta^v} \mathcal{J}_1 &= -\log p(\mathcal{S}|F^v) \\ &= -\sum_{s_{ij}\in\mathcal{S}} \left(s_{ij}\boldsymbol{f}_i^{v\top}\boldsymbol{f}_j^l - \log(1+\exp(\boldsymbol{f}_i^{v\top}\boldsymbol{f}_j^l))\right)\end{aligned} \tag{12}$$

where $\boldsymbol{f}_i^v$ is the semantic feature generated by *ImgNet*, and $\boldsymbol{f}_j^l$ is semantic feature from *LabNet*.

In common Hamming space, different from the traditional methods that employ pairwise similarity and iterative search hash codes, we guide the hash codes learning in *ImgNet* by utilizing the learned hash codes in *LabNet*. The hash layer of *ImgNet* is constrained to approach precise binary code $\{0,1\}^K$ by utilizing sigmoid function with cross-entropy loss. Since the activation function of hash layer in *LabNet* is $tanh(\cdot)$, the hash codes of *LabNet* need to adjust from $\boldsymbol{h}_i^l \in \{-1,1\}^K$ to $\boldsymbol{h}_i^{l'} \in \{0,1\}^K$ to match the $sigmiod(\cdot)$ activation function in *ImgNet*. The loss of hash codes in common Hamming space is defined as:

$$\min_{H^v,\theta^v} \mathcal{J}_2 = -\sum_{i=1}^{N}\left[\boldsymbol{h}_i^{l'}\log\sigma(\hat{\boldsymbol{y}}_i^v) + (1-\boldsymbol{h}_i^{l'})\log(1-\sigma(\hat{\boldsymbol{y}}_i^v))\right] \tag{13}$$

where $\hat{\boldsymbol{y}}_i^v$ is the output of *ImgNet*.

Therefore, the whole objective function of *ImgNet* is denoted as follow:

$$\min_{F^v,H^v,\theta^v} \mathcal{L}_{\mathrm{Img}} = \mathcal{J}_1 + \eta\mathcal{J}_2 \tag{14}$$

where η is the hyper-parameter to balance the two loss function terms.

3.4 Training Strategy

LabNet takes advantage of all label information to generate semantic features and hash codes. However, the learned semantic features and hash codes in *LabNet* may not match well with the corresponding semantic features and hash codes to be learned in *ImgNet* at the beginning. Therefore, we should exploit the strategy of alternative training to reconstruct the optimal semantic features and hash codes in semantic space and Hamming space, respectively.

Specifically, we first randomly initialize *LabNet* and train it until $\mathcal{L}_{\mathrm{lab}}$ reaches convergence. Then, utilizing the obtained semantic features and hash codes in *LabNet*, we supervise the *ImgNet* training in semantic space and Hamming space, respectively. Next, we initialize the semantic features and hash codes of *LabNet* with the resulting semantic feature and hash codes in *ImgNet* generated from the second step. Finally, repeating such training procedure for *LabNet* and *ImgNet* until convergence.

Algorithm 1 outlines the whole leaning algorithm in detail. It is noted that we learn all network parameters by utilizing stochastic gradient descent (SGD) with a back-propagation (BP) algorithm, which is also widely used in existing deep learning methods.

Algorithm 1 The learning algorithm for our DSEH

Input: Image set X , Label set L
Output: Parameters θ^v of *ImgNet*, Optimal code matrix B
Initialization
Initialize network parameters θ^l, θ^v.
hyper-parameters: α, β, γ, and η.
learning rate: μ.
mini-batch size: $N^l = 32$, $N^v = 128$.
maximum iteration number: t^l,t^v.
repeat
  for t^l iteration **do**
    Update θ^l by BP algorithm:
    $\theta^l \leftarrow \theta^l - \mu \cdot \nabla_{\theta^l}\frac{1}{n}(\mathcal{L}_{\mathrm{lab}})$
  end for
  Update the parameter $\boldsymbol{h}_i^l$ by $\boldsymbol{h}_i^l = sign(\boldsymbol{h}_i^l)$
  Update the parameter $\boldsymbol{h}_i^{l'}$ by adjusting $\boldsymbol{h}_i^l \in \{-1,1\}^K$ to $\in \{0,1\}^K$
  for t^v iteration **do**
    Update θ^v by BP algorithm:
    $\theta^v \leftarrow \theta^v - \mu \cdot \nabla_{\theta^v}\frac{1}{n}(\mathcal{L}_{\mathrm{img}})$
  end for
  Update the parameter $\boldsymbol{h}_i^v$, $\boldsymbol{h}_i^l$ by $\boldsymbol{h}_i^v = sign(\hat{\boldsymbol{y}}_i^v)$, $\boldsymbol{h}_i^l = sign(\hat{\boldsymbol{y}}_i^v)$
  Update the parameter B by $B = H^v$
until convergence

4 Experiments

4.1 Datasets and Settings

The experiments are conducted on three benchmark image retrieval datasets: NUS-WIDE [Chua *et al.*, 2009], ImageNet [Russakovsky *et al.*, 2015], and MS-COCO [Lin *et al.*, 2014].

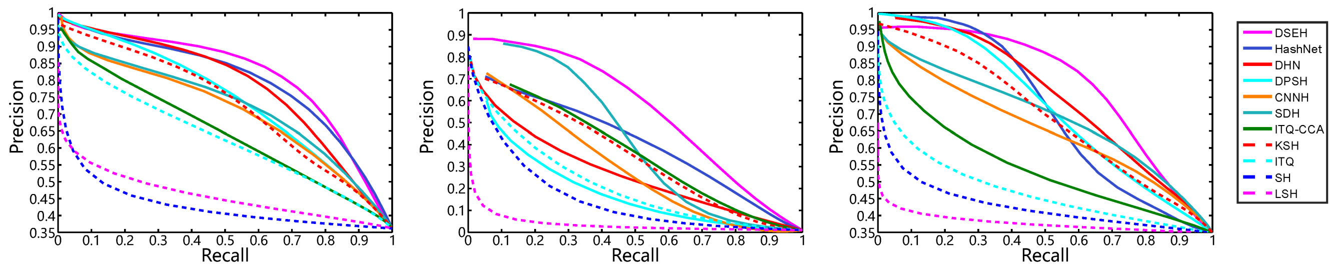
- ***NUS-WIDE*** dataset is a multi-label image dataset, which contains $269,648$ images with 81 ground truth concepts. We follow similar experimental protocols as DPSH [Li *et al.*, 2015] and use the subset of $195,834$ images that are associated with the 21 most frequent concepts, where each concept contains at least $5,000$ images. We randomly select 100 images per class as the query set, and 500 images per class as the training set.
- ***ImageNet*** dataset is a benchmark image dataset for Large Scale Visual Recognition Challenge (ILSVRC 2015), containing over 1.2M images. It is a single-label dataset, where each image is labeled by one of $1,000$ categories. We randomly select 100 categories, and randomly select 50 images per class as the query set, 100 images per class as the training set.
- ***MS-COCO*** dataset is an image recognition, segmentation and caption dataset which contains $82,783$ training images and $40,504$ validation images. It is a multi-label dataset labeled by 80 categories. After pruning images without category information, we obtain $122,218$ images and randomly sample $5,000$ images as queries, $10,000$ images as training points.

We evaluate the retrieval quality using three evaluation metrics: Mean Average Precision (MAP), Precision-Recall curves, and Precision curves with respect to the number of top returned results. With the same training and test set, all methods were tested under the same conditions. Given a query, the ground truth is defined as: if a result shares at least one common concept with the query, it is relevant; otherwise it is irrelevant.

We compare our method with ten classical or state-of-art hashing methods, including unsupervised methods

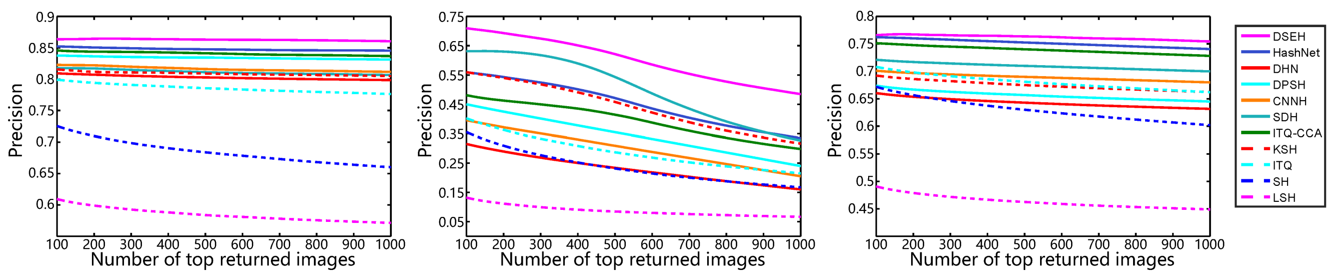
Method	NUS-WIDE				ImageNet				MS-COCO			
	16 bits	32 bits	48 bits	64 bits	16 bits	32 bits	48 bits	64 bits	16 bits	32 bits	48 bits	64 bits
DSEH	**0.7119**	**0.7312**	**0.7372**	**0.7422**	**0.5278**	**0.6137**	**0.6397**	**0.6548**	**0.5897**	**0.6048**	**0.6133**	**0.6188**
HashNet	0.7007	0.7275	0.7301	0.7374	0.3260	0.4563	0.5018	0.5270	0.5600	0.5850	0.5989	0.6056
DHN	0.6512	0.6611	0.6675	0.6741	0.1838	0.2344	0.2375	0.2564	0.5353	0.5456	0.5486	0.5555
DPSH	0.6902	0.7049	0.7130	0.7158	0.2730	0.2841	0.3111	0.3242	0.5618	0.5774	0.5857	0.5901
CNNH	0.6573	0.6601	0.6716	0.6781	0.2488	0.3047	0.3263	0.3387	0.5115	0.5232	0.5283	0.5328
SDH	0.6488	0.6703	0.6811	0.6857	0.3687	0.4292	0.4446	0.4600	0.5312	0.5632	0.5634	0.5741
ITQ-CCA	0.6125	0.6472	0.6655	0.6766	0.2312	0.4061	0.4316	0.4568	0.5418	0.5658	0.5704	0.5715
KSH	0.6404	0.6636	0.6689	0.6731	0.3064	0.3874	0.4006	0.4168	0.5496	0.5574	0.5628	0.5688
ITQ	0.5715	0.5876	0.5910	0.5985	0.1668	0.2452	0.2929	0.3184	0.4834	0.4993	0.5111	0.5153
SH	0.4459	0.4504	0.4342	0.4244	0.1194	0.1776	0.2143	0.2335	0.4494	0.4400	0.4397	0.4316
LSH	0.4624	0.4431	0.4433	0.4816	0.0278	0.0526	0.0720	0.0966	0.3718	0.3807	0.3945	0.4119

Table 1: Mean Average Precision(MAP) of Hamming Ranking on three benchmark datasets.



(a) NUS-WIDE (b) ImageNet (c) MS-COCO

Figure 3: Precision-recall curves @ 32bits of our method and comparison methods on three benchmark datasets.



(a) NUS-WIDE (b) ImageNet (c) MS-COCO

Figure 4: Precision w.r.t. top returned samples curves @ 32bits of our method and comparison methods on three benchmark datasets.

LSH [Gionis *et al.*, 1999], **SH** [Weiss *et al.*, 2009], **ITQ** [Gong *et al.*, 2013], supervised shallow methods **KSH** [Liu *et al.*, 2012], **ITQ-CCA** [Gong *et al.*, 2013], **SDH** [Shen *et al.*, 2015], and deep supervised methods **CNNH** [Xia *et al.*, 2014], **DPSH** [Li *et al.*, 2015], **DHN** [Zhu *et al.*, 2016], **HashNet** [Cao *et al.*, 2017].

For fair comparison, we extract $4,096$-dimensional deep features by CNN-F [Chatfield *et al.*, 2014] model which is retrained on ImageNet dataset. We construct *ImgNet* to reserve first seven layers same with those in CNN-F followed $fc8$ with 512 nodes for semantic layer and K nodes for hash layer, *i.e.*, $(I \rightarrow CNNF \rightarrow 512 \rightarrow K)$. *LabNet* is initialized randomly and constructed as $(L \rightarrow 4096 \rightarrow 512 \rightarrow K \rightarrow c)$, which contains c nodes for total class labels.

Since the semantic layer and hash layer are trained from scratch, we set its learning rate 10 times of the ones for the other layers. The learning rate is chosen from 10^{-2} to 10^{-6} with a validation set. The batch size of *LabNet* and *ImgNet* are set to 32 and 128 respectively. Since the semantic correlation of ImageNet is sparse, we set the values in similarity matrix as $\mathcal{S} \in \{0, 5\}$. For the hyper-parameters in *LabNet*, we conduct cross-validation to search α and γ from 10^{-3} to 10^{2}, and search β from 10^{-6} to 10^{-1}. We find that the optimal result can be obtained when $\alpha = \gamma = 1$, and $\beta = 0.005$. Then we search from 10^{-3} to 10^{2} and discover $\eta = 1$ is the best for *ImgNet*. It is noted that the parameter searching operations are performed with the searching step set to 5. Our model is implemented on **TensorFlow** [Abadi *et al.*, 2016] on a server with two NVIDIA TITAN X GPUs.

4.2 Results and Discussions

Table 1 shows the results of different hashing methods on three benchmark datasets when the code length is 16, 32, 48, and 64 bits respectively. Fig. 3 and Fig. 4 show the Precision-Recall curves and Precision curves respectively for different methods on the code length of 32 bits.

On two multi-label datasets NUS-WIDE and MS-COCO, DSEH substantially outperforms all the compared baseline methods. Besides, almost all deep hashing methods outper-

Method	NUS-WIDE			ImageNet			MS-COCO		
	n_h	map_l	map_i	n_h	map_l	map_i	n_h	map_l	map_i
DSEH	2008	**0.9664**	**0.7312**	100	**1.0000**	**0.6137**	1907	**0.8276**	**0.6048**
DSEH-S	1959	0.9633	0.7208	100	1.0000	0.5988	1933	0.8260	0.5907
DSEH-SS	1164	0.9322	0.7013	98	0.9825	0.5681	1220	0.7452	0.5237
DSEH-L	1036	0.9607	0.7251	100	1.0000	0.6070	802	0.8199	0.5915
DSEH-A	1684	0.9558	0.7234	100	1.0000	0.5576	1574	0.8134	0.5850

Table 2: The results of ablation study @ 32bits of our DSEH.

form the traditional hashing baselines, which highlights the benefit of feature learning by deep networks that more discriminative representation can be obtained. Compared with other deep methods which utilize similarity pairs, DSEH achieves a substantial increase in average MAP at different code lengths. All the results shown in Table 1, Fig. 3 and Fig. 4 illustrate the superiority of our method. One reason may be that instead of utilizing similarity pairs information roughly, DESH exploring label information to generate semantic feature is very effective to generate more sufficient semantic information and thus produce more discriminative hash codes. Another reason is that sufficient semantic information obtained from *LabNet* can be retained completely and thus supervise *ImgNet* effectively when training *ImgNet* with the supervised information on the semantic level and hash codes level.

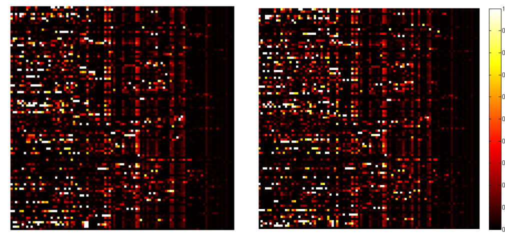
On ImageNet dataset which is annotated with single label. DHN, DPSH, and CNNH achieve under-performing results compared with the shallow baseline SDH, which demonstrates that network learning capacity can be dropped on single-label dataset because of the imbalance of pairs similarity. CNNH generates undiscriminating hash codes only under the supervision of pairwise similarity matrix. By adjusting the weight of similarity correlation, HashNet outperforms other baselines, which shows that adjusting weight can only alleviate influence of the data imbalance. The proposed DSEH significantly outperforms all other baselines. Compared with the state-of-the-art HashNet, we achieve about 34.50% increase in average MAP at different code lengths on this imbalanced dataset. It means that the proposed semantic feature learning and supervision to hashing learning can solve the issue of data imbalance in single-label dataset and thus hash codes can be generated more discriminative.

4.3 Empirical Analysis

Two different experiment settings are designed additionally to analyse the proposed method.

Visualization of Semantic Features: We visualize the semantic features generated by *LabNet* and *ImgNet* on NUS-WIDE at 32 bits in Fig. 5 (for convenience, 100 points are sampled and encapsulated by PCA [Wold *et al.*, 1987]). We observe that the semantic features of *LabNet* are abundant, indicating that the semantic information of labels is effectively exploited. Furthermore, the semantic features of *ImgNet* are similar to those in *LabNet*, inferring that *ImgNet* is well supervised in the common semantic space.

Ablation Study: We investigate the variants of DSEH on the three datasets. **DSEH-S** denotes that *ImgNet* without supervision on semantic layer from *LabNet*. **DSEH-SS** refers to that both *LabNet* and *ImgNet* without semantic supervision.



(a) *LabNet* (b) *ImgNet*

Figure 5: The visualization of semantic features.

DSEH-L denotes that *LabNet* drops direct label supervision. **DSEH-A** refers to that *LabNet* and *ImgNet* are trained only once without alternating manner.

Tabel 2 shows the average results of 10 runs of DSEH variants, where n_h is the total number of hash codes generated from *LabNet*, map_l is the MAP of retrieving labels with hash codes generated by *LabNet*, and map_i is the MAP of retrieving images with the hash codes generated by *ImgNet*. DSEH outperforms all of its variants, which shows the effectiveness of each module. DSEH-SS achieves the worst performance, the main reason of which is that semantic supervision plays a very important role in the proposed framework. It is noted that the higher n_h is, the more diverse hash codes can be generated. DSEH-L reduces n_h dramatically, illustrating that more semantic information can be maintained by adding label supervision to the proposed method.

5 Conclusion

In this paper, we proposed a novel deep hashing method, namely DSEH, for image retrieval, which consists of *LabNet* and *ImgNet*. The *LabNet* is explored to discover abundant semantic correlation and generate accurate hash codes. Meanwhile, the *ImgNet* is jointly constrained with the supervision information from common semantic space and common Hamming space for generating similarity-preserving yet discriminative hash codes. Extensive experiments conducted on three widely-used datasets demonstrate that our proposed method significantly outperforms many state-of-the-art hashing approaches, including both traditional and deep learning-based ones.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61572388 and Grant 61703327, and in part by the Key R&D Program-The Key Industry Innovation Chain of Shaanxi under Grant 2017ZDCXL-GY-05-04-02 and Grant 2017ZDCXL-GY-05-04-02.

References

[Abadi *et al.*, 2016] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[Cao *et al.*, 2017] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Philip S Yu. Hashnet: Deep learning to hash by continuation. *arXiv preprint arXiv:1702.00758*, 2017.

[Chatfield *et al.*, 2014] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014.

[Chua *et al.*, 2009] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. Nus-wide: a real-world web image database from national university of singapore. In *Proceedings of the ACM international conference on image and video retrieval*, page 48. ACM, 2009.

[Deng *et al.*, 2014] Cheng Deng, Rongrong Ji, Dacheng Tao, Xinbo Gao, and Xuelong Li. Weakly supervised multi-graph learning for robust image reranking. *IEEE transactions on multimedia*, 16(3):785–795, 2014.

[Deng *et al.*, 2015a] Cheng Deng, Huiru Deng, Xianglong Liu, and Yuan Yuan. Adaptive multi-bit quantization for hashing. *Neurocomputing*, 151:319–326, 2015.

[Deng *et al.*, 2015b] Cheng Deng, Xianglong Liu, Yadong Mu, and Jie Li. Large-scale multi-task image labeling with adaptive relevance discovery and feature hashing. *Signal Processing*, 112:137–145, 2015.

[Deng *et al.*, 2016] Cheng Deng, Xu Tang, Junchi Yan, Wei Liu, and Xinbo Gao. Discriminative dictionary learning with common label alignment for cross-modal retrieval. *IEEE Transactions on Multimedia*, 18(2):208–218, 2016.

[Deng *et al.*, 2018] Cheng Deng, Zhaojia Chen, Xianglong Liu, Xinbo Gao, and Dacheng Tao. Triplet-based deep hashing network for cross-modal retrieval. *IEEE TIP*, 2018.

[Gionis *et al.*, 1999] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. Similarity search in high dimensions via hashing. In *VLDB*, volume 99, pages 518–529, 1999.

[Gong *et al.*, 2013] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE TPAMI*, 35(12):2916–2929, 2013.

[Kulis and Darrell, 2009] Brian Kulis and Trevor Darrell. Learning to hash with binary reconstructive embeddings. In *Advances in neural information processing systems*, pages 1042–1050, 2009.

[Lai *et al.*, 2015] Hanjiang Lai, Yan Pan, Ye Liu, and Shuicheng Yan. Simultaneous feature learning and hash coding with deep neural networks. In *CVPR*, pages 3270–3278, 2015.

[Li *et al.*, 2015] Wu-Jun Li, Sheng Wang, and Wang-Cheng Kang. Feature learning based deep supervised hashing with pairwise labels. *arXiv preprint arXiv:1511.03855*, 2015.

[Li *et al.*, 2018] Chao Li, Cheng Deng, Ning Li, Wei Liu, Xinbo Gao, and Dacheng Tao. Self-supervised adversarial hashing networks for cross-modal retrieval. *arXiv preprint arXiv:1804.01223*, 2018.

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

[Liu *et al.*, 2011] Wei Liu, Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. Hashing with graphs. In *ICML*, pages 1–8. Citeseer, 2011.

[Liu *et al.*, 2012] Wei Liu, Jun Wang, Rongrong Ji, Yu-Gang Jiang, and Shih-Fu Chang. Supervised hashing with kernels. In *CVPR*, pages 2074–2081. IEEE, 2012.

[Liu *et al.*, 2016a] Haomiao Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. Deep supervised hashing for fast image retrieval. In *CVPR*, pages 2064–2072, 2016.

[Liu *et al.*, 2016b] Hong Liu, Rongrong Ji, Yongjian Wu, and Wei Liu. Towards optimal binary code learning via ordinal embedding. In *AAAI*, pages 1258–1265, 2016.

[Russakovsky *et al.*, 2015] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[Shen *et al.*, 2015] Fumin Shen, Chunhua Shen, Wei Liu, and Heng Tao Shen. Supervised discrete hashing. In *CVPR*, pages 37–45, 2015.

[Weiss *et al.*, 2009] Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. In *Advances in neural information processing systems*, pages 1753–1760, 2009.

[Wold *et al.*, 1987] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[Xia *et al.*, 2014] Rongkai Xia, Yan Pan, Hanjiang Lai, Cong Liu, and Shuicheng Yan. Supervised hashing for image retrieval via image representation learning. In *AAAI*, volume 1, pages 2156–2162, 2014.

[Yang *et al.*, 2017] Erkun Yang, Cheng Deng, Wei Liu, Xianglong Liu, Dacheng Tao, and Xinbo Gao. Pairwise relationship guided deep hashing for cross-modal retrieval. In *AAAI*, pages 1618–1625, 2017.

[Yang *et al.*, 2018] Erkun Yang, Cheng Deng, Chao Li, Wei Liu, Jie Li, and Dacheng Tao. Shared predictive cross-modal deep quantization. *IEEE TNNLS*, 2018.

[Zhu *et al.*, 2016] Han Zhu, Mingsheng Long, Jianmin Wang, and Yue Cao. Deep hashing network for efficient similarity retrieval. In *AAAI*, pages 2415–2421, 2016.