Towards Joint Multiply Semantics Hashing for Visual Search

Yunbo Wang[1,2] and Zhenan Sun[1,2]

[1] Institute of Automation, Chinese Academy of Sciences
[2] University of Chinese Academy of Sciences, Beijing, 100190, China
wang.yuenbo@gmail.com; znsun@nlpr.ia.ac.cn

Abstract. With the rapid growth of visual data on the web, deep hashing has shown enormous potential in preserving semantic similarity for visual search. Currently, most of the existing hashing methods employ pairwise or triplet-wise constraint to obtain the semantic similarity or relatively similarity among binary codes. However, some potential semantic context cannot be fully exploited, resulting in a suboptimal visual search. In this paper, we propose a novel deep hashing method, termed Joint Multiply Semantics Hashing (JMSH), to learn discriminative yet compact binary codes. In our approach, We jointly learn multiply semantic information to perform feature learning and hash coding. To be specific, the semantic information includes the pairwise semantic similarity between binary codes, the pointwise binary codes semantics and the pointwise visual feature semantics. Meanwhile, three different loss functions are designed to train the JMSH model. Extensive experiments show that the proposed JMSH yields state-of-the-art retrieval performance on representative image retrieval benchmarks.

Keywords: Deep Hashing · Binary codes · Multiply semantics · Visual search.

1 Introduction

With the explosive growth of image or video on the web, it is highly desirable that the data should be organized and indexed efficiently and accurately. As an approximate nearest neighbor (ANN) search technique, hashing [3,15,24,25] has shown superior potentials for dealing with large-scale visual data, which has received increasing attention in both the academia and industry. Generally, hashing employs a set of hashing functions to transform each data into compact binary codes, meanwhile retaining the semantic similarity of original data. Due to the encouraging efficiency in both search speed and storage [3,22], more and more hashing methods are proposed for visual retrieval tasks recently [2,26,29–31].

Generally, hashing methods could be divided into two main categories based on the type of hash functions: data-independent hashing [3, 10, 20] and data-dependent hashing (also known as learning-based hashing) [8, 23, 29]. Data-independent hashing methods always require long codes to achieve satisfying performance, while data-dependent hashing methods are prone to learning more compact binary codes by utilizing a batch of training data. In this paper, we focus on learning-based hashing with the application in visual search [21].

A fruitful of learning-based hashing methods have been designed for efficient ANN search, where the efficiency comes from the compact binary codes that are orders of magnitude smaller than high-dimensional feature descriptors. Based on the generated binary codes, the similarity between the query and the database can be efficiently computed. Meanwhile, the storage cost can be distinctly decreased. According to whether the supervision information is available, the learning-based hashing can be roughly grouped into unsupervised and supervised approaches. In contrast to unsupervised hashing [4,15,17,27] where no supervision information is provided, supervised hashing mainly leverages supervision information (e.g., pointwise semantic labels, pairwise similarity affinity) to perform hash learning. The supervised approaches have obtained better accuracy in real-world visual search. Some representative works include Minimal Loss Hashing [19], Supervised Discrete Hashing [22], Fast Supervised Discrete Hashing [18]. Recently, some approaches [2,13,29,32,33] have shown that convolutional neural network (CNN) [6,9] can be used as nonlinear hash functions to learn end-to-end feature representations and binary codes, achieving state-of-the-art results on public datasets.

The first proposed deep hashing work is Convolutional Neural Network Hashing (CNNH) [28], which adopts the well-known architecture in [9] to learn discriminative and compact binary codes with a pairwise constraint. CNNH consists of two stages to learn the feature representations and binary codes. Nevertheless, the feature representations cannot make feedback to hash coding and it cannot fully show the efficiency of CNN in hash learning. On the basis of CNNH, Network In Network Hashing (DNNH) [11] integrates image representations and hash coding in a unified framework. Besides, DNNH employs a triplet-based ranking constraint to maximize the margin between a similar pair and dissimilar pair, and it designs a divide-and-encode module to reduce the redundancy among binary codes. Furthermore, Deep Hashing Network (DHN) [33] is a representative pairwise deep hashing work in a unified framework. It employs a cross-entropy loss to enforce similar(dissimilar) pairs to have small(large) hamming distance and formally controls the pointwise quantization error by a designed smooth surrogate of the l_1-norm. To better control quantization error, HashNet [1] proposes a continuous scale strategy to approximately approach the discrete binary codes, and takes into consideration class imbalance to obtain small(large) hamming distance between data pair. DPH [2] also takes into consideration class imbalance for supervised hashing, and integrates the prior information into getting binary codes. Other typical deep hashing methods can be found in [2,12,15,16].

Among these methods above, they generally construct data pairs' similarity affinity as the ground truth for supervised hash learning. Specifically, the similarity is defined as 1 if two samples share at least one label information, and otherwise -1, then they employ the defined similarity affinity to obtain similarity-preserving binary codes in Hamming space. However, the defined similarity affinity fails to employ the high-level semantic information offered by label information, and the generated binary codes cannot show the high-level semantics. In addition, although existing hashing works perform feature learning and hash coding in an end-to-end way, they little make effort about extracting discriminative feature, as well as the effect on binary codes.

In this paper, we propose a joint multiply semantics hashing approach to address the above challenges. Specifically, we jointly learn three semantic properties to generate discriminative yet compact binary codes, including preserving the semantic similarity between a pair of binary codes, guaranteeing the pointwise codes' high-level semantics and learning the semantic visual feature.

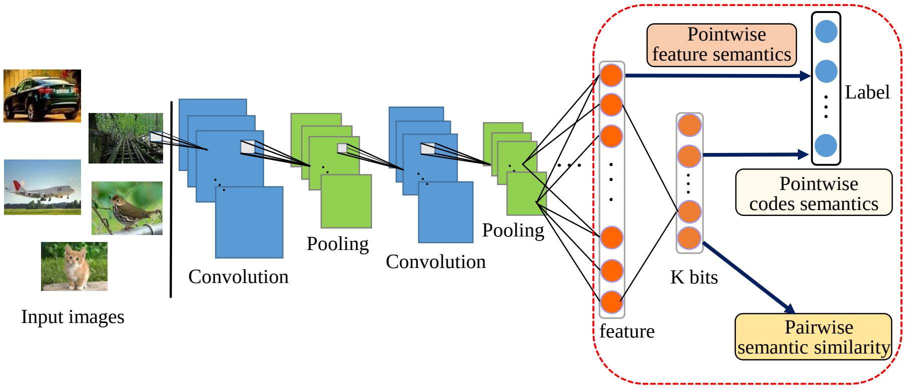

Input images
Convolution
Pooling
Convolution
Pooling
feature
K bits
Pointwise feature semantics
Label
Pointwise codes semantics
Pairwise semantic similarity

Fig. 1: An overview of the proposed deep hashing termed JMSH, which accepts image pairs as its input. In this framework, The AlexNet network is employed for extracting image feature, followed by a hashing layer with K neural units, which transforms the feature into K-bit binary codes. For each pair of binary codes, we utilize their similarity affinity to preserve the semantic similarity in Hamming space. Besides, we attempt to employ the label information to exploit the pointwise codes and feature semantic property.

2 Joint Multiply Semantics Hashing

Learning-based hashing has become an important research topic in multimedia retrieval, which trades off efficacy from efficiency. In this section, we will introduce our proposed joint multiply semantics hashing approach. The framework of the JMSH is shown in Figure 1, which accepts paired images as the input and processes them through the deep feature learning and hash coding. In this framework, we learn multiple semantic properties, including the pairwise codes semantic similarity, the pointwise codes semantics and the pointwise feature semantics.

2.1 Problem Formulation

Given a training set of N points $\boldsymbol{I} = \{\boldsymbol{I}_i\}_{i=1}^N$, the goal of learning-based hashing is to learn a set of hashing functions to encode each data point $\boldsymbol{I}_i$ into a compact K-bit binary code $\boldsymbol{B} = \{\boldsymbol{b}_i\}^{N \times K}$, $\boldsymbol{b}_i \in \{-1, 1\}^K$. The corresponding label matrix is denoted as $T = \{\boldsymbol{t}_i\}_i^N \in R^{\ N \times C}$ and C denotes the number of classes. The term $\boldsymbol{t}_{im}$ is the m-th element of $\boldsymbol{t}_i$ and $\boldsymbol{t}_{im} = 1$ if $\boldsymbol{I}_i$ is from class m, and otherwise $\boldsymbol{t}_{im} = 0$. Then, existing hashing generally denotes the paired similarity $s_{ij} = 1$ if two samples share at least one class label, and otherwise $s_{ij} = -1$ [23].

For training sample pairs $\{(I_i, I_j, s_{ij}), s_{ij} \in \boldsymbol{S}\}$, the discrete binary codes should preserve their similarity in Hamming space. Although several different objective functions can be leveraged to achieve this goal, the widely-used one is to leverage the inner product of two binary codes [2, 23] to approximate the discrete semantic similarity. Specifically, for a pair of codes $\boldsymbol{b}_i$ and $\boldsymbol{b}_j$, the close relationship between their Hamming distance $D_H(\boldsymbol{b}_i, \boldsymbol{b}_i)$ and their inner product $\boldsymbol{b}_i^T \cdot \boldsymbol{b}_j$ can be described as: $D_H(\boldsymbol{b}_i, \boldsymbol{b}_j) = \frac{1}{2}(k - \boldsymbol{b}_i^T \cdot \boldsymbol{b}_j)$. Given the pairwise similarity relationship S_{ij}, the maximum posterior estimation of binary codes can be described as:

$$p(\boldsymbol{B}|\boldsymbol{S}) \propto p(\boldsymbol{S}|\boldsymbol{B})p(\boldsymbol{B}) = \sum_{s_{ij}} p(s_{ij}|\boldsymbol{b}_i, \boldsymbol{b}_j)p(\boldsymbol{b}_i, \boldsymbol{b}_j), \tag{1}$$

where $p(\boldsymbol{S}|\boldsymbol{B})$ is the likelihood function; $p(\boldsymbol{B})$ is the prior distribution. For each pair of sample, $p(s_{ij}|\boldsymbol{b}_i, \boldsymbol{b}_j)$ is the conditional probability of similarity s_{ij} given a pairwise binary codes $(\boldsymbol{b}_i, \boldsymbol{b}_j)$. In particular, $p(s_{ij}|\boldsymbol{b}_i, \boldsymbol{b}_j)$ can be defined as follows:

$$p(s_{ij}|\boldsymbol{b}_i, \boldsymbol{b}_j) = \begin{cases} \sigma(\theta_{ij}), & s_{ij} = 1 \\ 1 - \sigma(\theta_{ij}), & s_{ij} = 0 \end{cases} \tag{2}$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the $sigmiod(\cdot)$ function; $\theta_{ij} = \eta \boldsymbol{b}_i^T \cdot \boldsymbol{b}_j$, and η is used to balance the saturation of $\sigma(x)$ in terms of different length of binary codes [1]. We can observe that the meaning of Equation 2 is highly consistent with the Hamming distance $d_H(\boldsymbol{b}_i, \boldsymbol{b}_j)$.

2.2 The Pairwise Semantic Similarity

Since deep learning [10] based hashing methods have shown superior performance over the traditional handcrafted feature [2], we construct an end-to-end framework based on Convolutional Neural Network to simultaneously perform feature learning and hash coding. In order to have a fair comparison with other deep hashing methods, we choose the widely-used AlexNet [10] as our basic network. The CNN model consists of 5 convolutional layers and 2 fully connected layers for extracting image feature $\boldsymbol{f}_i$. The hashing layer followed the connected layers is designed to encode $\boldsymbol{f}_i$ into binary codes $\boldsymbol{b}_i$. Specifically, the binary codes can be obtained by following formula:

$$\boldsymbol{b}_i = sign(\boldsymbol{W}_h \cdot \boldsymbol{f}_i), \tag{3}$$

where $\boldsymbol{W}_h$ is the weight of hashing layer and we omit its bias term for simplicity; $sign(\cdot)$ is the sign function, $sign(x) = 1$ if $x > 0$, and otherwise $sign(x) = -1$.

By taking the negative log-likelihood of the Equation 2, we can get the following optimization problem:

$$min \sum_{s_{ij}} (log(1 + e^{\theta_{ij}}) - s_{ij}\theta_{ij}), \tag{4}$$

It is easy to find that the above optimization problem can make the Hamming distance between two similar points as small as possible, and simultaneously make the Hamming distance between two dissimilar points as large as possible. This exactly matches the goal of supervised hashing with pairwise labels.

Due to the binary discrete constraint $\boldsymbol{b}_i \in \{-1, 1\}^K$, it is hard to optimize the Equation 4. As in existing hashing methods [16, 33], continuous relaxation is applied to the binary constraints. Meanwhile, we resort to l_2 regularizer to narrow the gap between the relaxation term and its corresponding binary codes:

$$min\, \mathcal{L}_{pair} = \sum_{s_{ij}} (log(1 + e^{\Omega_{ij}}) - s_{ij}\Omega_{ij}) + \frac{\alpha}{2} \sum_{i}^{N} ||\boldsymbol{h}_i - \boldsymbol{b}_i||^2 + \frac{1}{2} \|\boldsymbol{W}_h\|_F^2, \tag{5}$$

where $\boldsymbol{h}_i = \boldsymbol{W}_h \boldsymbol{f}_i$; $\Omega_{ij} = \eta \boldsymbol{h}_{\boldsymbol{i}}^T \cdot \boldsymbol{h}_{\boldsymbol{j}}$; $\| \cdot \|_F$ denotes the Frobenius norm.

2.3 The Pointwise Semantics

The label information offers rich high-level semantics of a raw image. The above similarity learning only employs the course similarity affinity for hash coding, resulting in the generated binary codes failing to show the rich semantic property of an image. Existing hashing methods make less research to exploit the relationship between the generated binary codes and label information.

To obtain specific-semantics binary codes, we attempt to reconstruct the label information by the generated binary codes:

$$min\, \mathcal{L}_b = \frac{1}{2} \sum_{i}^{N} ||\boldsymbol{W}_{\boldsymbol{b}} \boldsymbol{b}_i - \boldsymbol{t}_i||_2^2 + \frac{1}{2} \|\boldsymbol{W}_b\|_F^2, \tag{6}$$

where $\boldsymbol{W}_b$ is a line projection matrix, and $\boldsymbol{W}_{\boldsymbol{b}} \boldsymbol{b}_i$ denotes the reconstructed label information. Due to $\boldsymbol{t}_i \in \{0, 1\}^C$, we input the $\boldsymbol{W}_{\boldsymbol{b}} \boldsymbol{b}_i$ into the $sigmiod(\cdot)$ function to obtain approximated 0 or 1.

By Equation 6, we establish a non-linear relationship to link the binary codes and its corresponding label information, and the final binary codes can show the high-level semantic property of an image.

Since the proposed hashing method performs feature extracting and hash coding in an end-to-end way, the discriminative ability of feature inevitably makes an effect on the quality of hash coding. Although Most of existing deep hashing approaches simultaneously perform feature extracting and hash coding in a unified framework, they do nothing on how to extracting discriminative feature. As our above analysis, the image label provides supervised information for mining semantic structures in images.

In this paper, in order to make the feature have more discriminative power, we intentionally build a semantic relationship between feature representations and its label information:

$$min\ \mathcal{L}_f = \frac{1}{2}\sum_{i}^{N} ||\boldsymbol{W_f}\boldsymbol{f}_i - \boldsymbol{t}_i||^2 + \frac{1}{2}\|\boldsymbol{W}_f\|_F^2, \tag{7}$$

where $\boldsymbol{W}_f$ is a line projection matrix, and $\boldsymbol{W_f}\boldsymbol{f}_i$ denotes the predicted label information. Noting that we input $\boldsymbol{W_f}\boldsymbol{f}_i$ into the $sigmiod(\cdot)$ to obtain approximated 0 or 1. By the Equation 7, the feature is characteristic of the semantic property of an image, and the final feature representations have more discriminative power.

2.4 Joint Optimization

The proposed framework simultenously perform feature learning and hash coding, the final objective of the proposed JMSH is formulated as follow:

$$\begin{aligned} &min\,\mathcal{L}_{pair} + \beta_1\mathcal{L}_b + \beta_2\mathcal{L}_f \\ &\quad= \sum_{s_{ij}}(log(1+e^{\Omega_{ij}}) - s_{ij}\Omega_{ij}) + \frac{\alpha}{2}\sum_{i}^{N}||\boldsymbol{h}_i - \boldsymbol{b}_i||^2 + \frac{1}{2}\|\boldsymbol{W}_h\|_F^2 \\ &\quad+ \frac{\beta_1}{2}(\sum_{i}^{N}||\boldsymbol{W_b}\boldsymbol{h}_i - \boldsymbol{t}_i||^2 + \|\boldsymbol{W}_b\|_F^2) + \frac{\beta_2}{2}(\sum_{i}^{N}||\boldsymbol{W_f}\boldsymbol{f}_i - \boldsymbol{t}_i||^2 + \|\boldsymbol{W}_f\|_F^2). \end{aligned} \tag{8}$$

By the above formula, we can obtain the discriminative yet compact binary code in terms of learning multiple semantic properties. Learning a discriminative feature is conducive to obtain compact binary codes, and learning specific-semantics binary codes would improve the quality of binary codes. In Optimization, we adopt the stochastic gradient descent algorithm to update all these above parameters until convergence.

3 Experiments and Analysis

To evaluate the effectiveness of the proposed JMSH, extensive experiments are conducted on two benchmarks against the state-of-the-art hashing methods.

3.1 Datasets

CIFAR-10 is a benchmark image dataset for similarity retrieval, consisting of 60,000 color images. Each image belongs to one of the ten categories, and the size of each image is 32×32. Following the setting in [33], we sample 100 images per

class as the query set. For the unsupervised methods, all the rest of the images are used as the training set. For the supervised methods, 5,000 images (500 images per class) are further selected from the rest of the images for training.
NUS-WIDE is a public web image dataset downloaded from Flickr.com, and it contains nearly 270,000 images with one or multiple labels of 81 semantic concepts. Following the setting in HashNet [1], the subset of 195,834 images that are associated with the 21 most frequent concepts are used, where each concept consists of at least 5,000 images. We sample 100 images per class as the query set. For the unsupervised methods, all the rest of the images are used for training. For the supervised methods, 500 images per class are further selected from the rest images for training.

3.2 Experimental Setting and Protocols

As in standard evaluation protocol in [1, 2, 16], the similarity information for hash learning and for ground-truth evaluation is based on image class labels: if images i and j share at least one label, they are similar and $s_{ij} = 1$; otherwise, they are dissimilar and $s_{ij} = 0$. In addition, to avoid the effect caused by a class-imbalance problem between similar and dissimilar similarity information, we empirically set the weight of the similar pair as the the ratio between the number of dissimilar pairs and the number of similar pairs in image batch.

For the traditional hashing methods, each image is represented by a 4096-dim deep feature extracted from AlexNet [9] as the input. For the deep hashing methods, the raw image pixels are used as input. All deep methods adopt the AlexNet [7] as its basic architecture. In the JMSH, we fine-tune the front five convolutional layers and two fully-connected layers copied from the AlexNet model pre-trained on ImageNet2012 and train the semantic hashing layer. As the hashing layer is trained from scratch, we set its learning rate to be 10 times that of the lower layers. The initial learning rate is set to 10^{-5} and the weight decay parameter is 0.0005. The mini-batch size is fixed to be 200 and the input image is normalized to 256×256. For the hyper-parameters α, β_1 and β_2, we first fix $\beta_1 = 0$ and $\beta_2 = 0$, we conduct cross-validation to search α from 10^1 to 10^{-4}. We find that the optimal result can be obtained when setting α to be 10^{-1}. Then we search β_1 and β_2 from 10^1 to 10^{-5}, and we find the result is optimal when setting β_1 and β_2 to be 10^{-2} and 10^{-3}, respectively.

We compare retrieval performance of the **JMSH** with the classical state-of-the-art hashing methods, including the traditional hashing and deep hashing. The former includes **LSH** [3], **SH** [27], **ITQ** [5], **KSH** [10], **FastH** [14] and **SDH** [22]. The latter includes **DNNH** [11], **DHN** [33], **HashNet** [1] and **DPH** [2], where most of these methods obtains similarity-preserving binary codes according to the pairwise similarity affinity, such as DPH, HashNet, DHN, FastH and SH.

In the evaluation, several metrics are adopted to measure the quantitative performance. All methods are evaluated with four lengths of binary codes (8-bit, 16-bit, 24-bit and 32-bit), and under four standard evaluation metrics: Mean Average Precision (**MAP**), Precision-Recall curves (**PR**) and Precision curves within Hamming distance 2 (**P@H $\leq$ 2**). For fair comparisons, all methods use identical training and test sets, which are sampled from the dataset.

Table 1: Mean Average Precision (MAP) of Hamming Ranking for Different Number of Bits on Two Image Datasets.

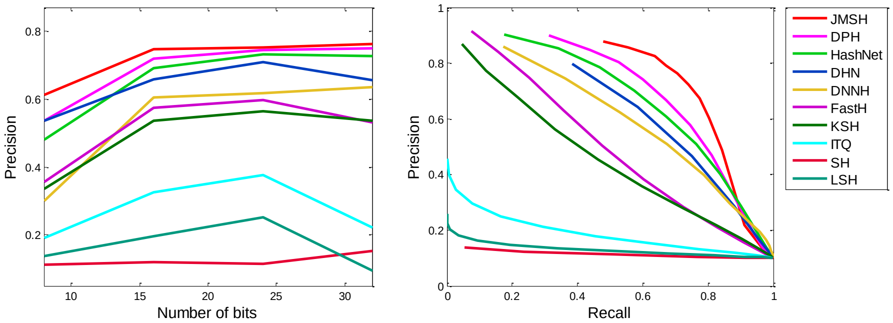
Method	CIFAR-10				NUS-WIDE			
	8 bits	16 bits	24 bits	32 bits	8 bits	16 bits	24 bits	32 bits
LSH [3]	0.1280	0.1368	0.1474	0.1637	0.1658	0.1867	0.2127	0.2494
SH [27]	0.1200	0.1254	0.1215	0.1277	0.1684	0.1694	0.1653	0.1765
ITQ [5]	0.1834	0.1997	0.2035	0.2087	0.2649	0.3142	0.3289	0.3407
KSH [10]	0.3860	0.4551	0.4701	0.4914	0.4696	0.5564	0.5684	0.5855
FastH [14]	0.4190	0.5006	0.5353	0.5436	0.5054	0.5962	0.6257	0.6386
SDH [22]	0.3192	0.5026	0.5318	0.5458	0.3608	0.5876	0.6080	0.6212
DNNH [11]	0.5561	0.6041	0.5876	0.5857	0.6121	0.6456	0.6574	0.6586
DHN [33]	0.5918	0.6554	0.6586	0.6601	0.6713	0.6823	0.6835	0.6871
HashNet [1]	0.6568	0.6925	0.7234	0.7401	0.6772	0.7001	0.7122	0.7239
DPH [2]	0.6672	0.6922	0.7243	0.7448	0.6852	0.7121	0.7199	0.7265
JMSH	**0.6962**	**0.7214**	**0.7326**	**0.7454**	**0.6916**	**0.7221**	**0.7316**	**0.7328**

3.3 Results and Discussions

Table 1 shows the MAP scores for different lengths of binary code on the CIFAR-10 and NUS-WIDE dataset, respectively. It is observed that our method constantly outperforms the baselines, including traditional hashing methods with CNN feature and deep learning based hashing methods.

Specifically, on the CIFAR-10 dataset, we can achieve an average MAP absolute increase of 24.88% compared to the traditional hashing method SDH [22] for different lengths of binary codes, and achieve an average MAP absolute increase of 2.04% and 1.65% compared to the state-of-the-art deep hashing methods HashNet [1] and DPH [2], respectively. For the NUS-WIDE dataset, the proposed JMSH shows a certain MAP improvement over these baselines, and the specific average MAP absolute increase can be up to 1.61% and 0.46% compared to the state-of-the-art hashing HashNet and DPH, respectively. The reason is that this dataset has in total of 21 class concepts and the structure information is more complicated among data pairs. Combined with the above analysis, the proposed JMSH show better results compared to the current hashing, the reason is that existing hashing methods mainly employ the pairwise similarity affinity to obtain similarity-preserving binary codes, overlooking the rich semantic information offered by label information. However, in the proposed JMSH, we further integrate the high-level semantic label information into the feature learning and binary codes learning, and improve the quality of the final binary codes.

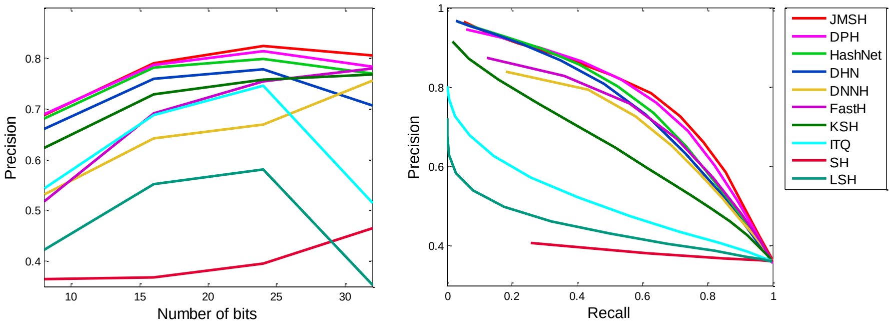
The performance in terms of Precision within Hamming radius 2 (P@H=2) is very important for efficient retrieval with binary codes since such Hamming ranking only requires O(1) time for each query. As shown in Figures 2-3(a), JMSH consistently achieves the best precision on two datasets. With the length of code becoming longer, P@H=2 of JMSH can still show a decreasing tendency. This validates that the JMSH can learn more compact binary codes than these baselines. As using longer codes, the Hamming space will become sparse and few data points fall within the Hamming ball with radius 2. This is why most hashing methods achieve the best accuracy with moderate code lengths.



(a) Precision within Hamming radius 2

(b) Precision-recall curve @ 16

Fig. 2: Comparative evaluations on the CIFAR-10 dataset. (a) Precision curves within Hamming distance 2; (b) Precision-recall curves with 16 bits.



(a) Precision within Hamming radius 2

(b) Precision-recall curve @ 16

Fig. 3: Comparative evaluations on the NUS-WIDE dataset. (a) Precision curves within Hamming distance 2; (b) Precision-recall curves with 16 bits.

The retrieval performance in terms of Precision-Recall curves (PR) is shown in Figures 2-3(b), respectively. It is clear that the JMSH shows a certain improvement compared to these comparison methods. Specifically, in low or high recall ratio, our method obtains a higher precision, which is desirable for precision-first practical retrieval systems. on the CIFAR-10, it shows a relatively higher initial recall over these baselines, and the reason is that the JMSH can put more similar pairs into the Hamming ball with low radius r, where the r increases from the minimum of 1 to the maximum of K (the code length). These obtained best results benefit from two components. First, We integrate the high-level semantics into the learning of binary codes. Second, We further learn the discriminative feature with the help of label information, and it is conducive to generate compact binary codes.

Table 2: Comparison of different loss terms in terms of MAP scores @ 16-bit binary codes.

	$\mathcal{L}_{pair}$	$\mathcal{L}_{pair} + \mathcal{L}_b$	$\mathcal{L}_{pair} + \mathcal{L}_f$	$\mathcal{L}_{pair} + \mathcal{L}_b + \mathcal{L}_f$
CIFAR-10	0.6986	0.7162	0.7059	0.7214
NUS-WIDE	0.7018	0.7155	0.7062	0.7221

3.4 Empirical Analysis

Table 2 reports the MAP scores of JMSH on two datasets about different loss functions. Each loss is corresponding to learning a semantic component, and reflects their individual effect in the objective function. The pairwise similarity learning loss $\mathcal{L}_{pair}$ is used to generate similarity-preserving binary codes in Hamming space; the pointwise codes semantics learning loss $\mathcal{L}_b$ explores the specific-semantics binary codes for enhancing the robust; the pointwise visual feature learning loss $\mathcal{L}_f$ facilitates the discriminative power of feature representations, and improves the quality of binary codes. It is observed that the three semantics learning can promote each other, generating the optimal binary codes for improving search performance.

4 Conclusion

This paper studies deep learning-based hashing approaches by learning multiply semantic properties to support efficient and effective visual search. The proposed deep hashing method, i.e., JMSH, can generate more compact binary codes based on three components: (1) learning the pairwise codes semantic similarity; (2) exploiting the pointwise codes high-level semantic property; (3) extracting more discriminative visual feature in an end-to-end framework. Extensive experimental results have shown the effectiveness of the proposed JMSH on two widely-used image retrieval datasets, compared with the state-of-the- art methods. In the future, we further exploit the multiple semantics learning on cross-modal datasets, improving cross-modal retrieval accuracy.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (Grant No. U1836217, 61427811, 61573360) and the National Key Research and Development Program of China (Grant No. 2017YFC0821602, 2016YFB1001000).

References

1. Cao, Z., Long, M., Wang, J., Philip, S.Y.: Hashnet: Deep learning to hash by continuation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5609–5618 (2017)
2. Cao, Z., Sun, Z., Long, M., Wang, J., Yu, P.S.: Deep priority hashing. In: Proceedings of the ACM Multimedia Conference on Multimedia. pp. 1653–1661. ACM (2018)
3. Gionis, A., Indyk, P., Motwani, R., et al.: Similarity search in high dimensions via hashing. In: Proceedings of International Conference on Very Large Data Bases. pp. 518–529 (1999)
4. Gong, Y., Lazebnik, S.: Iterative quantization: A procrustean approach to learning binary codes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 817–824 (2011)
5. Gong, Y., Lazebnik, S., Gordo, A., Perronnin, F.: Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(12), 2916–2929 (2013)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
7. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)
8. Jegou, H., Douze, M., Schmid, C.: Product quantization for nearest neighbor search. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(1), 117–128 (2011)
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. pp. 1097–1105 (2012)
10. Kulis, B., Grauman, K.: Kernelized locality-sensitive hashing. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(6), 1092–1104 (2012)
11. Lai, H., Pan, Y., Liu, Y., Yan, S.: Simultaneous feature learning and hash coding with deep neural networks. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 3270–3278 (2015)
12. Li, Q., Sun, Z., He, R., Tan, T.: Deep supervised discrete hashing. In: Advances in Neural Information Processing Systems. pp. 2482–2491 (2017)
13. Li, W.J., Wang, S., Kang, W.C.: Feature learning based deep supervised hashing with pairwise labels. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. pp. 1711–1717 (2016)
14. Lin, G., Shen, C., Shi, Q., Van den Hengel, A., Suter, D.: Fast supervised hashing with decision trees for high-dimensional data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1963–1970 (2014)
15. Lin, K., Lu, J., Chen, C.S., Zhou, J.: Learning compact binary descriptors with unsupervised deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1183–1192 (2016)
16. Liu, H., Wang, R., Shan, S., Chen, X.: Deep supervised hashing for fast image retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2064–2072 (2016)

17. Liu, W., Mu, C., Kumar, S., Chang, S.F.: Discrete graph hashing. In: Advances in Neural Information Processing Systems. pp. 3419–3427 (2014)
18. Luo, X., Nie, L., He, X., Wu, Y., Chen, Z.D., Xu, X.S.: Fast scalable supervised hashing. In: The International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 735–744 (2018)
19. Norouzi, M., Blei, D.M.: Minimal loss hashing for compact binary codes. In: Proceedings of the International Conference on Machine Learning. pp. 353–360 (2011)
20. Raginsky, M., Lazebnik, S.: Locality-sensitive binary codes from shift-invariant kernels. In: Advances in Neural Information Processing Systems. pp. 1509–1517 (2009)
21. Shen, F., Gao, X., Liu, L., Yang, Y., Shen, H.T.: Deep asymmetric pairwise hashing. In: Proceedings of the 25th ACM international conference on Multimedia. pp. 1522–1530. ACM (2017)
22. Shen, F., Shen, C., Liu, W., Tao Shen, H.: Supervised discrete hashing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 37–45 (2015)
23. Shen, F., Wei, L., Zhang, S., Yang, Y., Shen, H.T.: Learning binary codes for maximum inner product search. In: Proceedings of IEEE International Conference on Computer Vision. pp. 4148–4156. IEEE (2015)
24. Tang, J., Li, Z., Wang, M., Zhao, R.: Neighborhood discriminant hashing for large-scale image retrieval. IEEE Transactions on Image Processing **24**(9), 2827–2840 (2015)
25. Wang, J., Zhang, T., Sebe, N., Shen, H.T., et al.: A survey on learning to hash. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(4), 769–790 (2018)
26. Wang, Y., Liang, J., Cao, D., Sun, Z.: Local semantic-aware deep hashing with hamming-isometric quantization. IEEE Transactions on Image Processing **28**(6), 2665–2679 (2018)
27. Weiss, Y., Torralba, A., Fergus, R.: Spectral hashing. In: Advances in Neural Information Processing Systems. pp. 1753–1760 (2009)
28. Xia, R., Pan, Y., Lai, H., Liu, C., Yan, S.: Supervised hashing for image retrieval via image representation learning. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence. vol. 1, pp. 2156–2162 (2014)
29. Yang, H.F., Lin, K., Chen, C.S.: Supervised learning of semantics-preserving hash via deep convolutional neural networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(2), 437–451 (2018)
30. Zhang, D., Wang, J., Cai, D., Lu, J.: Self-taught hashing for fast similarity search. In: Proceedings of the ACM SIGIR conference on Research and Development in Information Retrieval. pp. 18–25 (2010)
31. Zhang, P., Zhang, W., Li, W.J., Guo, M.: Supervised hashing with latent factor models. In: Proceedings of ACM SIGIR conference on Research and development in information retrieval. pp. 173–182. ACM (2014)
32. Zhang, R., Lin, L., Zhang, R., Zuo, W., Zhang, L.: Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification. IEEE Transactions on Image Processing **24**(12), 4766–4779 (2015)
33. Zhu, H., Long, M., Wang, J., Cao, Y.: Deep hashing network for efficient similarity retrieval. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. pp. 2415–2421 (2016)