Deep Semantic-Preserving and Ranking-Based Hashing for Image Retrieval

Ting Yao,[†] **Fuchen Long,**[‡] **Tao Mei,**[†] **Yong Rui**[†]
[†]Microsoft Research, Beijing, China
[‡]University of Science and Technology of China, Hefei, China
{tiyao, tmei, yongrui}@microsoft.com, longfc.ustc@gmail.com

Abstract

Hashing techniques have been intensively investigated for large scale vision applications. Recent research has shown that leveraging supervised information can lead to high quality hashing. However, most existing supervised hashing methods only construct similarity-preserving hash codes. Observing that semantic structures carry complementary information, we propose the idea of co-training for hashing, by jointly learning projections from image representations to hash codes and classification. Specifically, a novel deep semantic-preserving and ranking-based hashing (DSRH) architecture is presented, which consists of three components: a deep CNN for learning image representations, a hash stream of a binary mapping layer by evenly dividing the learnt representations into multiple bags and encoding each bag into one hash bit, and a classification stream. Meanwhile, our model is learnt under two constraints at the top loss layer of hash stream: a triplet ranking loss and orthogonality constraint. The former aims to preserve the relative similarity ordering in the triplets, while the latter makes different hash bit as independent as possible. We have conducted experiments on CIFAR-10 and NUS-WIDE image benchmarks, demonstrating that our approach can provide superior image search accuracy than other state-of-the-art hashing techniques.

1 Introduction

The rapid development of Web 2.0 technologies has led to the surge of research activities in large scale visual search [Mei *et al.*, 2014]. One fundamental research problem is similarity search, i.e., nearest neighbor search, which attempts to identify similar instances according to a query example. The need to search for millions of visual examples in a high-dimensional feature space, however, makes the task computationally expensive and thus challenging.

Hashing techniques, one direction of the most well-known Approximate Nearest Neighbor (ANN) search methods, have received intensive research attention for its great efficiency in gigantic data. The basic idea of hashing is to construct a series of hash functions to map each example into a compact binary code, making the Hamming distances on similar examples minimized and simultaneously maximized on dissimilar examples. In the literature, there have been several techniques, including traditional hashing models based on hand-crafted features [Weiss *et al.*, 2008; Wang *et al.*, 2012] and deep models [Lai *et al.*, 2015; Liong *et al.*, 2015], being proposed for addressing the problem of hashing. The former seek hashing function on hand-crafted features, which separate the encoding of feature representations and their quantization to hash codes, resulting in sub-optimal solution. The latter jointly learn feature representations and projections from them to hash codes in a deep architecture. We are investigating in this paper how to design a deep architecture for hashing to characterize the relative similarity between images, meanwhile making the obtained hash bits as independent as possible.

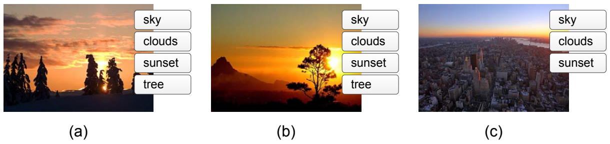


Figure 1: Three exemplary images. Both images in (a) and (b) are associated with four tags: “sky,” “clouds,” “sunset” and “tree.” The image in (c) is labeled with “sky,” “clouds” and “sunset.”

While existing hashing approaches are promising to measure similarity, the relationship between two images is more complex especially when images are with multiple semantic labels and is usually reflected by the number of common labels that two images have. Figure 1 shows three exemplary images. The two images in (a) and (b) are both associated with “sky,” “clouds,” “sunset” and “tree,” while the image in (c) is only relevant to “sky,” “clouds” and “sunset.” In this case, the hash codes of image in (a) should be closer in proximity to the image in (b) than the image in (c). Therefore, in practice, how to preserve semantic structures of the data in form of class labels is also essential to be further taken into account for hashing.

By consolidating the idea of co-training between hashing and preserving semantic structures, this paper presents a novel Deep Semantic-Preserving and Ranking-Based Hashing (DSRH) architecture, as illustrated in Figure 2. The input to our architecture is in the form of triplets, i.e., a query image, a similar image and a dissimilar image. A shared DCNN is then exploited to produce image representations, followed by importing into a hash stream for hash code encoding and a classification stream for measuring semantic structures. A triplet ranking loss is designed with orthogonality constraint to characterize relative similarities at the top of hash stream, while a classification error is formulated in classification stream. By jointly learning hash stream and classification stream, the generated hash codes are expected to better present semantic similarities between images.

The main contributions of this paper include:

- We propose a novel hashing architecture, which combines hash coding and classification for preserving not only relative similarity between images but also semantic structures on images.
- A triplet ranking loss with orthogonality constraint is exploited to optimize our architecture, making each hash bit as independent as possible.
- An extensive set of experiments on two widely used datasets demonstrate the advantages of our proposed model over several state-of-the-art hashing techniques.

2 Related Work

We briefly group related works into two categories: hand-crafted features based hashing and deep models for hashing.

The research on hand-crafted features based hashing has proceeded along three dimensions: unsupervised hashing, semi-supervised hashing and supervised hashing. Unsupervised hashing [Gionis *et al.*, 1999; Gong and Lazebnik, 2011] refers to the setting when the label information is not available. Locality Sensitive Hashing (LSH) [Gionis *et al.*, 1999] is one of the most well-known representative, which simply uses random linear projections to construct hash functions. Another effective method called Iterative Quantization (ITQ) [Gong and Lazebnik, 2011] was suggested for better quantization rather than random projections. Spectral Hashing (SH) in [Weiss *et al.*, 2008] was proposed to design compact binary codes by preserving the similarity between samples, which can be viewed as an extension of spectral clustering [Zelnik-manor and Perona, 2014]. Semi-supervised hashing approaches try to improve the quality of hash codes by leveraging the supervised information into learning procedure. For example, Wang *et al.* developed a Semi-Supervised Hashing (SSH) [Wang *et al.*, 2012] method which utilizes pairwise information on labeled samples to preserve relative similarity. In another work [Kim and Choi, 2011], Semi-Supervised Discriminant Hashing (SSDH) learns hash codes based on Fisher's discriminant analysis to maximize separability between labeled data in different classes while the unlabeled data are used for regularization. When all label information is available, we refer to the problem as supervised hashing. The representative in this category is Kernel-based Supervised Hashing (KSH) which was proposed by Liu *et al.* in [Liu *et al.*, 2012]. It maps the data to compact binary codes whose Hamming distances are minimized on similar pairs and simultaneously maximized on dissimilar pairs. In [Norouzi and Fleet, 2011], Norouzi *et al.* proposed Minimal Loss Hashing (MLH) method, which aims to learn similarity-preserving binary codes by exploiting pairwise relationship.

Inspired by recent advances in image representation using deep convolutional neural networks, a few deep architecture based hashing methods have been proposed for image retrieval. Semantic Hashing [Salakhutdinov and Hinton, 2009] is the first work to exploit deep learning techniques for hashing. It applies stacked Restricted Boltzmann Machine (RBM) to learn hash codes for visual search. Xia *et al.* proposed Convolutional Neural Networks Hashing (CNNH) [Xia *et al.*, 2014] to decompose the hash learning process into a stage of learning approximate hash codes followed by a deep-networks-based stage of simultaneously learning image features and hash functions. However, separating hashing learning into two stages may result in a sub-optimal solution. Later in [Lai *et al.*, 2015], Lai *et al.* proposed Network in Network Hashing (NINH) to combine feature learning and hash coding into one stage.

In summary, our approach belongs to deep architecture based hashing. The aforementioned approaches often focus on similarity-preserving learning for hashing. Our work in this paper contributes by studying not only preserving relative similarity between images, but also how image semantic supervision could be further leveraged for boosting hashing.

3 Deep Semantic-preserving and Ranking-based Hashing (DSRH)

In this section, we will present the proposed Deep Semantic-Preserving and Ranking-Based Hashing (DSRH) in details. Figure 2 illustrates an overview of our architecture, which consists of three components: a shared DCNN for learning image representations, a hash stream for encoding each image into hash codes and a classification stream for leveraging semantic supervision. Specifically, the hash stream is designed with multiple bags construction plus orthogonality constraint trained in a triplet-wise manner, while the classification stream reinforces the hash learning to preserve semantic structures on images. We will discuss the two streams in the Section 3.2 and 3.3, respectively.

3.1 Notation

Suppose that we have n images and each of them can be presented as X. The goal of image hashing is to learn a mapping $\mathcal{H} : X \rightarrow \{0,1\}^k$, such that an input image X will be encoded into a *k*-bit binary code $\mathcal{H}(X)$.

3.2 Hash Stream

The hash coding of each image is treated independently in point-wise hashing learning methods, regardless of the relationships of similar or dissimilar between images. More importantly, the relative similarity relations like "for query image X, it should be more similar to image X^+ than to image X^-," are reflected in the image class labels that image X and X^+ belong to the same class while image X^- comes from other categories. The utilization of these relative similarity relations has been proved to be effective in hash coding [Pan *et al.*, 2015; Lai *et al.*, 2015; Li *et al.*, 2014]. Inspired by the

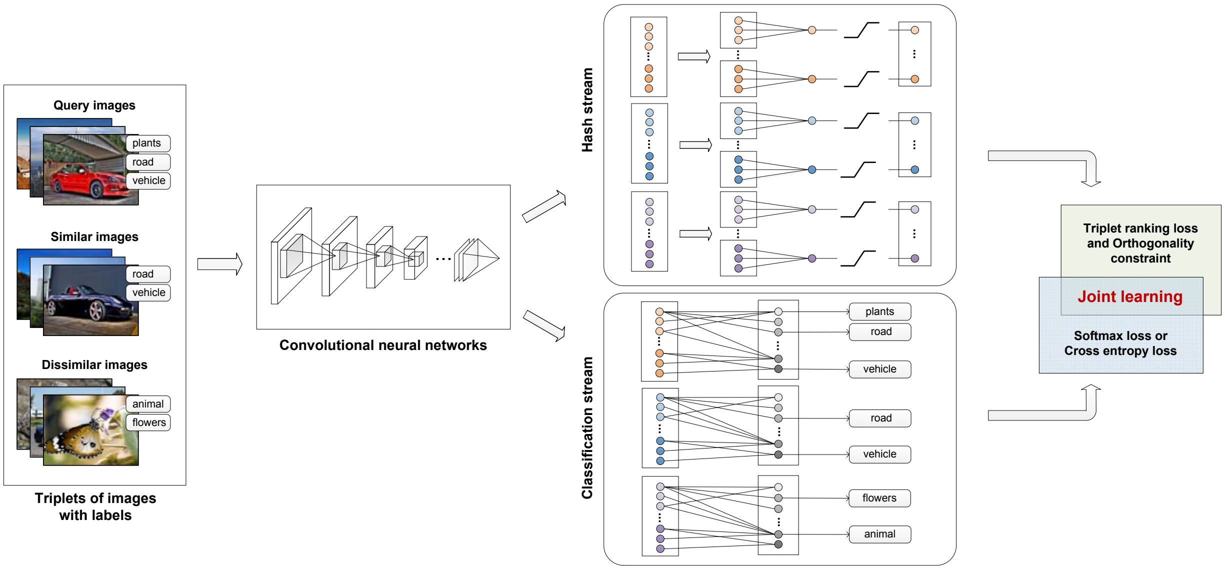


Figure 2: Deep Semantic-Preserving and Ranking-Based Hashing (DSRH) framework (better viewed in color). The input to DSRH architecture is in the form of image triplets. A shared deep convolutional neural networks is exploited for learning image representations, followed by two streams, i.e., hash stream and classification stream. Hash stream is to encode each image into hash codes by first dividing the learnt representations into multiple bags and then convert each bag into one hash bit. Triplet loss with orthogonality constraint is measured in hash stream. Classification stream is to characterize the semantic structures on image and softmax loss or cross entropy loss is computed for single label and multi-label classification, respectively. Both hash stream and classification stream are jointly learnt by minimizing two losses.

idea of preserving relative similarity in deep architecture [Lai *et al.*, 2015], we propose a hash stream with multiple bags construction plus orthogonality constraint learnt in a triplet-wise manner, which aims to preserve relative similarity as well as make each hash bit as independent as possible.

Triplet Ranking Loss

Specifically, we can easily obtain a set of triplets $\mathcal{T}$ based on image labels, where each tuple (X, X^+, X^-) consists of a query image X, a similar image X^+ and a dissimilar image X^-. To preserve the similarity relations in the triplets, we aim to learn a hash mapping $\mathcal{H}(\cdot)$ which makes the compact code $\mathcal{H}(X)$ more similar to $\mathcal{H}(X^+)$ than to $\mathcal{H}(X^-)$. Thus, the triplet ranking hinge loss is employed and defined as

$$\begin{aligned}&\hat{l}_{triplet}(\mathcal{H}(X), \mathcal{H}(X^+), \mathcal{H}(X^-))\\&= \max(0, 1 - \left\|\mathcal{H}(X) - \mathcal{H}(X^-)\right\|_H + \left\|\mathcal{H}(X) - \mathcal{H}(X^+)\right\|_H) \;,\\&\;s.t. \quad \mathcal{H}(X), \mathcal{H}(X^+), \mathcal{H}(X^-) \in \{0,1\}^k\end{aligned} \tag{1}$$

where $||\cdot||_H$ represents Hamming distance. For ease of optimization, natural relaxation tricks are utilized on Eq.(1) to change integer constraint to the range constraint and replace Hamming norm with l_2 norm. Then, the loss function is reformulated as

$$\begin{aligned}&l_{triplet}(\mathcal{H}(X), \mathcal{H}(X^+), \mathcal{H}(X^-))\\&= \max(0, 1 - \left\|\mathcal{H}(X) - \mathcal{H}(X^-)\right\|_2^2 + \left\|\mathcal{H}(X) - \mathcal{H}(X^+)\right\|_2^2) \;.\\&\;s.t. \quad \mathcal{H}(X), \mathcal{H}(X^+), \mathcal{H}(X^-) \in [0,1]^k\end{aligned} \tag{2}$$

The gradients in the back-propagation of the triplet ranking loss are computed as

$$\begin{aligned}\frac{\partial l_{triplet}}{\partial \mathbf{h}} &= \left(2\mathbf{h}^- - 2\mathbf{h}^+\right) \times I_{\|\mathbf{h}-\mathbf{h}^+\|_2^2 - \|\mathbf{h}-\mathbf{h}^-\|_2^2 + 1 > 0}\\ \frac{\partial l_{triplet}}{\partial \mathbf{h}^+} &= \left(2\mathbf{h}^+ - 2\mathbf{h}\right) \times I_{\|\mathbf{h}-\mathbf{h}^+\|_2^2 - \|\mathbf{h}-\mathbf{h}^-\|_2^2 + 1 > 0} \;,\\ \frac{\partial l_{triplet}}{\partial \mathbf{h}^-} &= \left(2\mathbf{h} - 2\mathbf{h}^-\right) \times I_{\|\mathbf{h}-\mathbf{h}^+\|_2^2 - \|\mathbf{h}-\mathbf{h}^-\|_2^2 + 1 > 0}\end{aligned} \tag{3}$$

where $\mathcal{H}(X), \mathcal{H}(X^+), \mathcal{H}(X^-)$ are represented as vector $\mathbf{h}, \mathbf{h}^+, \mathbf{h}^-$, respectively. The indicator function $I_{condition} = 1$ if *condition* is true; otherwise $I_{condition} = 0$.

Multiple Bags Construction plus Orthogonality Constraint

For hashing representation learning, compactness is a critical criterion to guarantee its performance in efficient similarity search. Given a certain small length of binary codes, the redundancy lies in different bits would badly affect its performance. By removing the redundancy, we can either incorporate more information in the same length of binary codes, or shorten the binary codes while maintaining the same amount of information. Thus to alleviate the redundancy problem, we develop multiple bags construction in our deep architecture. The multiple bags module has a unique construction which divides the input features into k bags firstly and then encodes each bag into one hash bit by a fully-connected layer. It aims to reduce the bit redundancy and the effectiveness has been proved in the hashing work [Lai *et al.*, 2015]. Moreover, an orthogonality constraint is further imposed at the top loss layer of hash stream to decorrelate different hash bit.

Let m and k denote the number of triplets in a batch and the number of output hash bits, respectively. After we obtain the matrix $\mathbf{H} \in [0,1]^{m\times k}$ of hash bits, a projection $\tilde{\mathbf{H}} = 2\mathbf{H} - \mathbf{1}$ is exploited to transform $\mathbf{H}$ to $\tilde{\mathbf{H}} \in [-1,1]^{m\times k}$. Thus, the loss function with orthogonality constraint is given by

$$\begin{aligned} & \min(l_{triplet}(\tilde{\mathbf{H}}^*, \tilde{\mathbf{H}}^+, \tilde{\mathbf{H}}^-)) \\ s.t. \quad & \tilde{\mathbf{H}}^*, \tilde{\mathbf{H}}^+, \tilde{\mathbf{H}}^- \in [-1,1]^{m\times k} \\ & \frac{1}{m}\tilde{\mathbf{H}}^T\tilde{\mathbf{H}} = \mathbf{I} \quad \tilde{\mathbf{H}} \in \{\tilde{\mathbf{H}}^*, \tilde{\mathbf{H}}^+, \tilde{\mathbf{H}}^-\} \end{aligned}, \tag{4}$$

where $\tilde{\mathbf{H}}^*, \tilde{\mathbf{H}}^+, \tilde{\mathbf{H}}^-$ represents the matrix of the approximate hash bits of query images, similar images and dissimilar images from all the triplets, respectively.

The orthogonality constraint in Eq.(4) makes the optimization difficult to be solved. To address this problem, the orthogonal constraint $\frac{1}{m}\tilde{\mathbf{H}}^T\tilde{\mathbf{H}} = \mathbf{I}$ can be relaxed by appending the converted soft penalty term to the objective function. Then, the final loss function can be rewritten as

$$\min(l_{triplet}(\tilde{\mathbf{H}}) + \lambda l_{orthogonal}(\tilde{\mathbf{H}})) \ , \tag{5}$$

where the hyper-parameter λ is the tradeoff parameter between triplet ranking loss and orthogonality constraint. Formally, the orthogonality constraint loss is

$$l_{orthogonal}(\tilde{\mathbf{H}}) = \frac{1}{3}\sum \left\| \frac{1}{m}\tilde{\mathbf{H}}^T\tilde{\mathbf{H}} - \mathbf{I} \right\|_F^2, \tag{6}$$

where $||\cdot||_F$ represents the Frobenius norm.

Therefore, the gradient of the orthogonality constraint loss respect to hash stream is computed by

$$\frac{\partial l_{orthogonal}}{\partial \tilde{\mathbf{H}}} = \frac{4}{3m}\tilde{\mathbf{H}}\left(\tilde{\mathbf{H}}^T\tilde{\mathbf{H}} - \mathbf{I}\right), \tag{7}$$

where $\tilde{\mathbf{H}} \in \{\tilde{\mathbf{H}}^*, \tilde{\mathbf{H}}^+, \tilde{\mathbf{H}}^-\}$.

3.3 Classification Stream

Image labels not only provide knowledge in classifying but also are useful supervised information for mining semantic structures in images. A valid question is how to leverage the semantic supervision into hashing and make the generated hash codes better reflecting semantic similarities between images. Specifically, we propose a co-training mechanism by combining hash stream and classification stream. In the classification stream, a classification error is measured and the whole architecture of the two streams are jointly learnt by minimizing triplet ranking loss in hash stream and classification loss in classification stream.

Softmax Optimization

For the single label classification, we use softmax optimization method. Given an input image x^i, the softmax loss is then formulated as

$$l_{softmax}(\theta) = -\frac{1}{n}\left[\sum_{i=1}^{n}\sum_{j=1}^{c} I_{(y^i=j)} log \frac{e^{\theta_j^T x^i}}{\sum_{l=1}^{c} e^{\theta_l^T x^i}}\right], \tag{8}$$

where θ denotes the parameters of our architecture and $y^i \in \{1, 2, ...c\}$ represents image label.

The gradient with respect to θ_j for optimization is

$$\frac{\partial l_{softmax}}{\partial \theta_j} = -\frac{1}{n}\sum_{i=1}^{n}\left[x^i(I_{(y^i=j|x^i)} - \hat{p}(y^i = j|x^i;\theta))\right], \tag{9}$$

where $\hat{p}(y^i = j|x^i;\theta)$ is the predicted probability:

$$\hat{p}(y^i = j|x^i;\theta) = \frac{e^{\theta_j^T x^i}}{\sum_{l=1}^{c} e^{\theta_l^T x^i}} \ . \tag{10}$$

Cross Entropy Optimization

If an image contains multiple labels, we refer to this problem as multi-label classification. Cross entropy loss is employed in this case. Similar to softmax loss, cross entropy loss is computed by

$$\begin{aligned} l_{CrossEntropy}(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c} & [p_i(\mathbf{y}_j^i = 1) log(\hat{p}_i(\hat{\mathbf{y}}_j^i = 1;\theta)) \\ & + (1 - p_i(\mathbf{y}_j^i = 1)) log(1 - \hat{p}_i(\hat{\mathbf{y}}_j^i = 1;\theta))] \end{aligned}, \tag{11}$$

where $\hat{p}_i$ is the predict probability which is the same as Eq.(10). $\mathbf{y}^i \in \{0,1\}^c$ is a binary label vector, where c is the number of labels.

3.4 Image Retrieval

After the optimization of DSRH, we can employ hash stream in the architecture to generate k-bit hash codes for each input image. In this procedure, an image X is first encoded into a k-dimension feature vector $\mathbf{h}$. Then, a quantization operation $\mathbf{b} = sign(\mathbf{h})$ is exploited to generate hash codes $\mathbf{b}$, where $sign(\mathbf{h})$ is a sign function on vector $\mathbf{h}$ with $sign(h_i) = 1$ if $h_i > 0$ and otherwise $sign(h_i) = 0$. Given a query image, the retrieval list of images is produced by sorting the hamming distances of hash codes between the query image and images in search pool.

4 Experiments

We conducted extensive evaluations of our proposed method on two image datasets, i.e., CIFAR-10[1], a tiny image collection and NUS-WIDE[2], a large scale web image dataset.

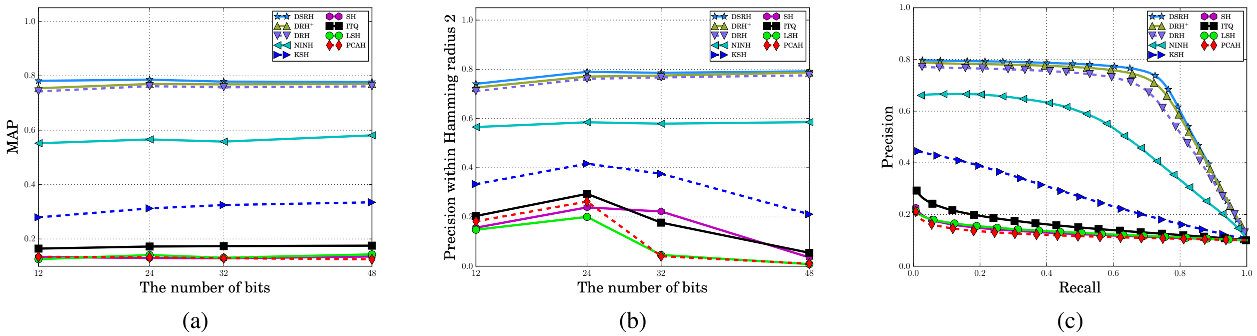
4.1 Dataset

The **CIFAR-10** dataset consists of 60,000 real world tiny images in 10 classes. Each class has 6,000 images in size 32 × 32. We randomly select 1,000 images (100 images per class) as the test query set. For the unsupervised setting, all the rest images are used as training samples. For the supervised setting, we additionally sample 500 images from each class in the training samples and constitute a subset of 5,000 labeled images for training.

The **NUS-WIDE** dataset contains 269,648 images collected from Flickr. Each of these images is associated with one or multiple labels in 81 semantic concepts. For a fair comparison, we follow the settings in [Lai *et al.*, 2015] to use the subset of images associated with 21 most frequent labels,

[1] http://www.cs.toronto.edu/ kriz/cifar.html

[2] http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm



(a)

(b)

(c)

Figure 3: Comparisons with state-of-the-art approaches on CIFAR-10 dataset. (a) Mean average precision (MAP) performance. (b) Precision within Hamming radius 2 using hash lookup. (c) Precision-Recall curves on 48 bits. For better viewing, please see original color pdf file.

where each label associates with at least 5,000 images. We resize each image to 256×256. Similar to the split in CIFAR-10, we randomly select 2,100 images (100 images per class) as the test query set. For the unsupervised setting, all the rest images are used as the training set. For the supervised setting, we uniformly sample 500 images from each class to construct a subset for training.

4.2 Experimental Settings

On both datasets, we utilize the 19-layer VGG [Simonyan and Zisserman, 2015] as our basic DCNN architecture. In between, the first 18 layers follow the exactly same architectures as VGG network and the number of neurons in the last fully-connected layer is set to $s \times k$, where s and k is the number of bags and hash bits, respectively. We empirically set $s = 30$ in all our experiments. The hyper-parameter λ is determined by using a validation set and set to 0.25 finally.

We implement the proposed method based on the open-source **Caffe** [Jia *et al.*, 2014] framework. In all experiments, our networks are trained by stochastic gradient descent with 0.9 momentum. The start learning rate is set to 0.01, and we decrease it to 10% after $5,000$ iterations on CIFAR-10 and after $20,000$ iterations on NUS-WIDE. The mini-batch size of images is 64. The weight decay parameter is 0.0002.

4.3 Protocols and Baseline Methods

We follow three evaluation protocols, i.e., mean average precision (MAP), hash lookup and precision-recall curve, which are widely used in [Gong and Lazebnik, 2011; Liu *et al.*, 2012; Wang *et al.*, 2012]. We compare the performances of our proposed model $DSRH$ with six hashing methods including five traditional models, i.e., PCA Hashing ($PCAH$), Locality Sensitive Hashing (LSH) [Gionis *et al.*, 1999], Spectral Hashing (SH) [Weiss *et al.*, 2008], Iterative Quantization (ITQ) [Gong and Lazebnik, 2011] and Supervised Hashing with Kernels (KSH) [Liu *et al.*, 2012], and one deep model, i.e., Network In Network Hashing ($NINH$) [Lai *et al.*, 2015]. Moreover, two slightly different settings of our $DSRH$ are named as DRH^+ and DRH, which only includes individual hash stream with and without orthogonality constraint, respectively.

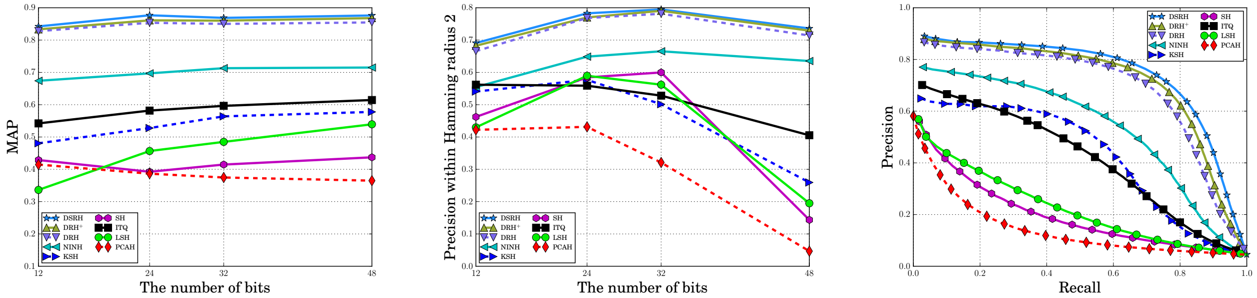
For $NINH$, DRH, DRH^+ and $DSRH$, the raw pixel images are set as input. For the other baseline methods, we use the 512-dimensional GIST vector for each image in CIFAR-10 and the output of 1000-way fc8 classification layer in Alexnet [Krizhevsky *et al.*, 2012] for NUS-WIDE.

4.4 Results on CIFAR-10 Dataset

Figure 3(a) shows the MAP performances of nine runs on CIFAR-10 dataset. Overall, the results across different number of hash bits consistently indicate that our $DSRH$ outperforms others. In particular, the MAP of $DSRH$ makes the relative improvement over the best traditional competitor KSH and deep model $NINH$ by 132.1%∼179.5% and 33.8%∼41.6%, respectively, which is so far the highest performance reported on CIFAR-10 dataset. There is a significant performance gap between the traditional and deep models. It is not surprising to see that DRH improves $NINH$ since DRH exploits a more powerful image representation brought by a deeper CNN. By additionally incorporating orthogonality constraint, DRH^+ exhibits better performance than DRH. Our $DSRH$ further improves DRH^+ with a relative increase of 1.6%∼3.6%, demonstrating the advantage of boosting hashing by preserving semantic structures through classification.

In the evaluation of hash lookup within Hamming radius 2 as shown in Figure 3(b), the precisions for most of the traditional methods drop when a longer size of hash codes is used (48 bits in our case). This is because the number of samples falling into a bucket decreases exponentially for longer sizes of hash codes. Therefore, for some query images, there are even no any neighbor in a Hamming ball of radius 2. Even in this case, the precision of our proposed $DSRH$ still has a very slight improvement from 78.57% of 32 bits to 79.17% of 48 bits, indicating fewer failed queries for $DSRH$.

We further detail the precision-recall curves in Figure 3(c). The results confirm the trends seen in Figure 3(a) and demonstrate performance improvements using the proposed $DSRH$ approach over other methods.



(a)

(b)

(c)

Figure 4: Comparisons with state-of-the-art approaches on NUS-WIDE dataset. (a) Mean average precision (MAP) performance. (b) Precision within Hamming radius 2 using hash lookup. (c) Precision-Recall curves on 48 bits. Better viewed in original color pdf file.

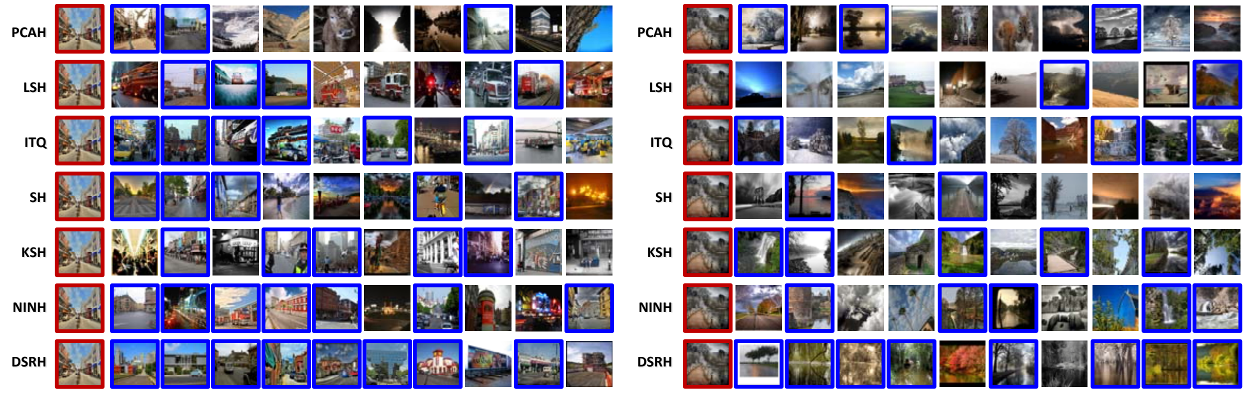


Figure 5: Examples showing the top 10 image retrieval results by different methods in response to two query images on NUS-WIDE dataset (better viewed in color). In each row, the first image with a red bounding box is the query image and the images whose annotations completely contain all the labels of the query image are regarded as excellent ones, which are enclosed in a blue bounding box.

4.5 Results on NUS-WIDE Dataset

Figure 4 shows the experimental results on NUS-WIDE dataset. MAP performance and precision with Hamming radius 2 using hash lookup are given in Figure 4(a) and (b), respectively. Our $DSRH$ model consistently outperforms others. In particular, the MAP performance and precision with Hamming radius 2 using hash lookup of $DSRH$ can achieve 87.61% and 73.59% with 48 hash bits, which make the improvement over the best competitor $NINH$ by 22.53% and 15.85%. Furthermore, $DSRH$, in comparison, is benefited from utilizing semantic supervision and thus shows a relative increase of 1.0%∼1.8% over DRH^+ in terms of MAP. Similar to the observations on CIFAR-10 dataset, the precisions of most methods decrease when increasing the size of hash codes to 48 bits and the drop in precision of our $DSRH$ is much less compared to others. Figure 4(c) shows the precision-recall curves and the results indicate that $DSRH$ constantly leads to better performance.

Figure 5 further illustrates the top ten image search results by different methods in response to two query images. We can see that the proposed $DSRH$ method achieves more satisfying results and retrieves eight "excellent images" in the returned top ten images. It is worth noticing that "excellent images" here refer to images whose annotations completely contain all the labels of the query image. As a result, the images retrieved by our $DSRH$ approach are more similar in semantics with the query image.

5 Conclusion and Discussion

In this paper, we have presented deep semantic-preserving and ranking-based hashing architecture which explores both relative similarity between images and semantic supervision on images. Particularly, given triplets of images with labels, we exploit a shared DCNN to learn image representation, followed by two streams, i.e., hash stream and classification stream. Hash stream aims to encode each image into hash codes by characterizing the relative similarities between images in a triplet and meanwhile making the generated hash codes as compact as possible, while classification stream is to preserve the semantic structures on images. Basically, utilizing only hash stream shows better performance than state-of-the-art hashing techniques on two image datasets. By jointly learning hash stream and classification stream to reinforce hashing, further improvements are consistently observed in the experiments.

Our future works are as follows. First, as our architecture is a co-training process, how the architecture performs on classification task will be further investigated and evaluated. Next, more in-depth studies of how to fuse the two streams in a principle way could be explored. Furthermore, how to apply our proposed architecture to video domain seems interesting.

References

[Gionis *et al.*, 1999] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Similarity search in high dimensions via hashing. In *Proceedings of International Conference on Very Large Data Bases (VLDB)*, 1999.

[Gong and Lazebnik, 2011] Yunchao Gong and Svetlana Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2011.

[Jia *et al.*, 2014] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross B. Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *CoRR*, 2014.

[Kim and Choi, 2011] Saehoon Kim and Seungjin Choi. Semi-supervised discriminant hashing. In *Proceedings of International Conference on Data Mining (ICDM)*, 2011.

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of Neural Information Processing Systems (NIPS)*, 2012.

[Lai *et al.*, 2015] Hanjiang Lai, Yan Pan, Ye Liu, and Shuicheng Yan. Simultaneous feature learning and hash coding with deep neural networks. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2015.

[Li *et al.*, 2014] Yinxiao Li, Yan Wang, Michael Case, Shih-Fu Chang, and Peter K. Allen. Real-time pose estimation of deformable objects using a volumetric approach. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[Liong *et al.*, 2015] Venice Erin Liong, Jiwen Lu, Gang Wang, Pierre Moulin, and Jie Zhou. Deep hashing for compact binary codes learning. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2015.

[Liu *et al.*, 2012] Wei Liu, Jun Wang, Rongrong Ji, Yu-Gang Jiang, and Shih-Fu Chang. Supervised hashing with kernels. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2012.

[Mei *et al.*, 2014] Tao Mei, Yong Rui, Shipeng Li, and Qi Tian. Multimedia search reranking: A literature survey. *ACM Computing Surveys*, 46(3), 2014.

[Norouzi and Fleet, 2011] Mohammad Norouzi and David J. Fleet. Minimal loss hashing for compact binary codes. In *Proceedings of International Conference on Machine Learning (ICML)*, 2011.

[Pan *et al.*, 2015] Yingwei Pan, Ting Yao, Houqiang Li, Chong-Wah Ngo, and Tao Mei. Semi-supervised hashing with semantic confidence for large scale visual search. In *Proceedings of ACM conference on Research and Development in Information Retrieval (SIGIR)*, 2015.

[Salakhutdinov and Hinton, 2009] R. Salakhutdinov and G.E. Hinton. Semantic hashing. *International Journal of Approximate Reasoning*, 50(7):969–978, 2009.

[Simonyan and Zisserman, 2015] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2015.

[Wang *et al.*, 2012] Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. Semi-supervised hashing for large scale search. *IEEE Trans, Pattern Anal. Mach. Intell (TPAMI)*, 34(12):2393–2406, 2012.

[Weiss *et al.*, 2008] Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. In *Proceedings of Neural Information Processing Systems (NIPS)*, 2008.

[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 *Proceedings of Association for the Advancement of Artificial Intelligence (AAAI)*, 2014.

[Zelnik-manor and Perona, 2014] Lihi Zelnik-manor and Pietro Perona. Self-tuning spectral clustering. In *Proceedings of Neural Information Processing Systems (NIPS)*, 2014.