Learning Multilevel Semantic Similarity for Large-Scale Multi-Label Image Retrieval

Ge Song
Nanjing University of Aeronautics and Astronautics
Nanjing, China
sunge@nuaa.edu.cn

Xiaoyang Tan
Nanjing University of Aeronautics and Astronautics
Nanjing, China
x.tan@nuaa.edu.cn

ABSTRACT

We present a novel Deep Supervised Hashing with code operation (DSOH) method for large-scale multi-label image retrieval. This approach is in contrast with existing methods in that we respect both the intention gap and the intrinsic multilevel similarity of multi-labels. Particularly, our method allows a user to simultaneously present multiple query images rather than a single one to better express her intention, and correspondingly a separate sub-network in our architecture is specifically designed to fuse the query intention represented by each single query. Furthermore, as in the training stage, each image is annotated with multiple labels to enrich its semantic representation, we propose a new margin-adaptive triplet loss to learn the fine-grained similarity structure of multi-labels, which is known to be hard to capture. The whole system is trained in an end-to-end manner, and our experimental results demonstrate that the proposed method is not only able to learn useful multilevel semantic similarity-preserving binary codes but also achieves state-of-the-art retrieval performance on three popular datasets.

CCS CONCEPTS

• **Information systems** → **Image search**;

KEYWORDS

Multi-Label Image Retrieval; Hashing; Deep Learning

ACM Reference Format:
Ge Song and Xiaoyang Tan. 2018. Learning Multilevel Semantic Similarity for Large-Scale Multi-Label Image Retrieval. In *Proceedings of 2018 International Conference on Multimedia Retrieval (ICMR '18).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3206025.3206027

1 INTRODUCTION

With the popularization of social media and the explosive growth of the web images, content-based image retrieval, which aims to search the images displaying the same objects categories or visual content as the query, has attracted increasing attention. The field has a wide range of application scenarios (e.g., product search, automatic image annotation) but is challenging as well. There are three types of difficulties commonly encountered in practice [32]: the first is known as semantic gap, which is originated from the limited representative of low-level visual features for high-level semantic concepts; the second one concerns about the storage and retrieval efficiency due to the large-scale data to be retrieved; last but not the least, the user's intention is usually hard to be precisely expressed as the expected visual content by a query at hand, which is known as intention gap in literature and is less studied compared to the previous two.

One of the most recent popular techniques to address these issues is the method of deep hashing [2, 13, 16, 18, 21, 28]. It combines the advantages of the feature learning capability of deep convolutional neural networks and the low storage and computational cost of the hashing method to map images into compact binary codes that preserve semantic similarity. In this sense, deep hashing methods address both the semantic gap and the computational efficiency issue of image retrieval but leave the third challenge, i.e., the intention gap, mostly untouched. As a matter of fact, it is almost unrealistic to rely only on a single image to accurately reflect the complex query intention of a user. In literatures, the intention gap is often tackled with multiple alternative query formations, e.g., sketch-based [20, 24, 26, 29] and image-text [7, 12]. Such alternatives serve to be the complement to the original query and allow the user to express her query intention in more flexible ways. Unfortunately, these methods may not always work, for example, under the situations when users are not good at sketching. In addition, they impose an extra burden to learning algorithms to handle different modal data.

To alleviate such problems and inspired by the previous work [12], we propose to leverage data from the same modal to augment query semantic, i.e., allowing a user to freely select multiple query images (e.g., a 'bicycle' image and 'person' image) to express more complex query intention (e.g., 'a person riding a bicycle'). Another way to narrow the intention gap is by exploiting semantic information of each image. Previous work [7] showed that there exists high consistency between users' query intention and human-annotated region-level captions, which means that the labels of query images have a close connection with the user's intention. Fortunately, nowadays many images are associated with more than one

ICMR '18, June 11–14, 2018, Yokohama, Japan

ACM ISBN 978-1-4503-5046-4/18/06…$15.00
https://doi.org/10.1145/3206025.3206027

tags (referred as multi-label) which provide a valuable rich semantic description of the corresponding image.

However, multi-label image retrieval is not straightforward, and query images with multiple labels may actually enlarge the intention gap, as the multiple labels would increase the entropy of user's intention over those labels. Another issue lies in the inherent complexity of the similarity between multi-label images, which involves the measurement of similarity not only at the image-level but the semantic (label) level as well. Many early deep hashing schemes [1, 3, 14, 15, 18–20, 22, 25] are designed for single-label images, which only concern about the preservation of simple binary semantic similarity (i.e., similar or dissimilar), and the Hamming distance between the learned codes cannot accurately reflect the hierarchical relevance in semantics.

Several more recent deep hashing methods [13, 21, 28, 31] have specifically devoted to the topic of multi-label image retrieval. One popular way for this is to explicitly model the similarity between multi-labels, for example, by forcing the codes of 'dog+person' and 'dog+cat' images as close as possible to those of 'dog' images. One side effect of imposing such constraints in the semantic space is that the learned hashing codes are prone to be too ambiguous to express fine-grained concepts, i.e., making the companying individual concepts (e.g., the person' and 'cat' in the previous example) less distinguishable in the representation space. Moreover, due to the imbalance and sparsity problem commonly encountered in multi-label image retrieval, the learned codes tend to be biased to only a few tags. This urges the need to construct more informative codes to respect the complex similarity structure involved in such conditions.

In this work, we introduce a novel deep supervised hashing with code operation (DSOH) for multi-label image retrieval. An overview of the proposed framework is illustrated in Fig. 1. To simultaneously deal with the intention gap problem and learn multilevel similarity of multi-label images, a specific sub-network in our approach is designed to fuse multiple codes at the semantic level, which is named code operation. In training phase, it is able to generate codes that contain rich semantic information to reflect the inherent complex similarity structure. For this, a new margin-adaptive triplet loss is proposed to learn the fine-grained similarity among images with multiple labels. This loss, along with the weighted cross-entropy classification loss, is employed to guide the end-to-end training of the model. While in the testing phase, this sub-network allows a user to present multiple images to express her intention better. Our main contributions are summarised as follows:

- A novel deep hashing method, named DSOH, is proposed to learn multilevel semantic similarity-preserving and operable binary codes for multi-label image retrieval. Unlike the previous hashing methods that ignore the *intention gap* problem in retrieval, a new Code Operation Network is designed to perform the 'union,' 'intersect' and 'substract' operations on semantic concepts of multiple query image codes, which effectively enables users to expand their query intention in a more flexible and friendly manner.
- A new margin-adaptive triplet loss is introduced to capture fine-grained multilevel similarity among images with multiple labels.
- Both standard multi-label retrieval task and complex semantic retrieval tasks are conducted using the proposed DSOH approach on three multi-label image datasets, and state-of-the-art retrieval performance is achieved.

The remainder of this paper is organized as follows. We first discuss related work in section 2. The DSOH model is detailed in section 3, and experimental results are given in section 4. The paper is concluded in section 5.

2 RELATED WORK

Recently several deep hashing schemes [13, 20, 21, 28, 31] have been proposed to learn efficient binary codes for multi-label image retrieval. DSRH (Deep Semantic Ranking based Hashing) [31] tries to learn hashing functions that preserve multilevel semantic similarity via optimizing an adaptively weighted triplet loss which aims to penalize undesired orders with different similarity degrees. IAH (Instance-Aware Hashing) [13] focuses on learning instance-aware hashing for multi-label images with weighted triplet loss functions, in which images are represented by multiple pieces of semantic binary codes corresponds to different categories. MSDH (Multi-label Supervised Deep Hashing) [21] uses multi-layer non-linear transformations as the hashing function and learns similarity via weighted pairwise loss, while DMSSPH (Deep Multilevel Semantic Similarity Preserving Hashing) [28] designs a pairwise loss with adaptive margins that imposes constraints on the distance between the learned codes.

Although most of these supervised hashing methods are successful in performing multi-label semantic retrieval to some extent, seldom of them have attempted to exploit the learn codes to express complex query intention. Closely related works are [7, 12, 23, 30]. Among them, [7] learns a joint embedding of visual and textual cues, allowing one to query the database using a text modifier in addition to editing the query image for semantic retrieval, and [30] introduces a memory-augmented Attribute Manipulation Network to manipulate image representation at the attribute level with a given set of attribute tags for fashion search. Both methods allow users to enrich their query intention by adjusting image representation with directly provided semantic information, the success which, however, heavily depends on the skill of users when using these tools.

3 THE PROPOSED APPROACH

In this section, we first give the notations used in this paper. And then the configurations of the proposed DSOH is detailed, including two sub-networks and the corresponding loss function.

3.1 The problem definition

Assume that we are given a set of N images $\{I_n\}_{n=1}^{N}$. Each image I_n is associated with C classes and its corresponding label l_n is denoted as a binary vector $\{0,1\}^C$. Each image is

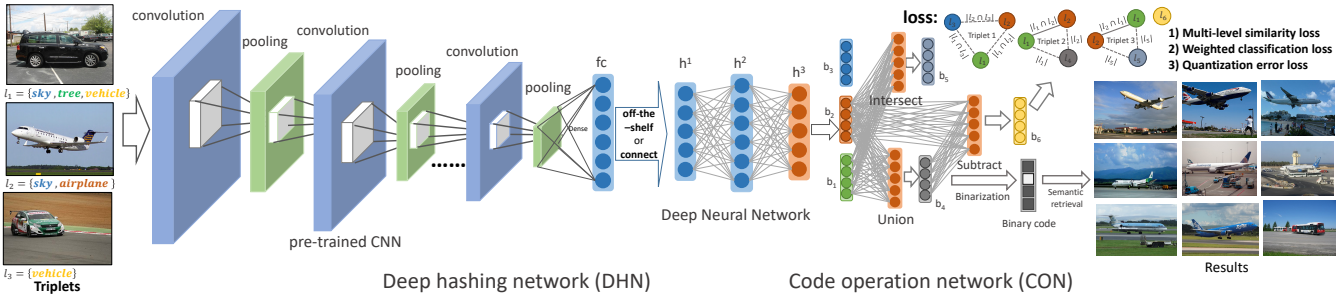


Figure 1: The architecture of the proposed deep supervised hashing network with code operation (DSOH), which includes two sub-networks, one deep hashing network (DHN), and one code operation network (CON). In the training phase, the hashing network takes as input a triplet images I_1,I_2,I_3 and their labels and outputs correspondingly three hashing codes b_1, b_2, b_3, which are then processed by the CON network with three types of code operations, i.e., union, substraction and intersection, respectively. The resulting codes can be used in the testing phase for multi-label image retrieval.

also represented by a D-dimensional visual features $x_i \in R^D$. Our goal is to learn a set of K ($K \ll D$) hashing functions $h_i(x), i = 1...K$, each of which maps a given visual feature x to a binary code $\{-1, 1\}^1$ and jointly they forms a K-dimensional hashing codes $h(x) = [h_1(x), h_2(x), \ldots, h_k(x)]$ which the semantic similarity among images is preserved.

3.2 Deep Hashing Network

Our deep hashing network (DHN) consists of a CNN-based feature learner and a fully-connect deep neural network (DNN). The CNN-based feature learner we adopted is the commonly-used CNN model such as VGG and ResNet pre-trained on the ImageNet. The structure of DNN with M layers is detailed as follows,

$$\begin{aligned} h^1 &= \tanh(w^1 x + bias^1) \\ h^m &= \tanh(w^{m-1} x + bias^1) \\ h^M &= \tanh(w^M h^{M-1} + bias^M) \end{aligned} \tag{1}$$

where x denotes feature vectors from the outputs of the last layer of CNN (i.e., $x = f_{CNN}(I)$, where f_{CNN} is a CNN network.), h^m and w^m respectively being the output and the weights of the m-th layer. The hashing codes can be obtained with a thresholding process on the output of DNN:

$$b = sign(h(x)) = \begin{cases} 1 \ , h(x) > 0 \\ -1 \ , h(x) \leq 0 \end{cases} \tag{2}$$

3.3 Code Operation Network

The learned codes of previous semantic hashing methods are independent and cannot be used collaboratively to express more complex queries. Therefore a code operation network (CON) is designed based on the output of the deep hashing network to manipulate them at the semantic level. Three types of code operation are supported in our work, i.e., union, substraction and intersection, all of which are the pairwise operator which takes in a pair of hashing codes and transforms them into a new one. For two codes $h(x_1)$ and $h(x_2)$, a union operator f_u fuses them into one $h_{x_{1\vee 2}} = f_u(h(x_1), h(x_2))$ with label $l_u = l_1 \oplus l_2 = l_1 \cup l_2$; a intersect f_t to fuse them into codes $h_{x_{1\wedge 2}} = f_t(h(x_1), h(x_2))$ with label $l_t = l_1 \otimes l_2$, defined as $l_1 \cap l_2$; and a substractor f_s substracts $h(x_1)$ with $h(x_2)$ to give $h_{x_{1\setminus 2}} = f_s(h(x_1), h(x_2))$ with label $l_s = l_1 \ominus l_2$, which is equal to $l_1 - l_1 \cap l_2$ if $|l_1| > |l_1 \cap l_2|$, and $l_s = l_1$ otherwise.

Union and Intersection Operation. Given two hashing codes $h(x_1)$, $h(x_2)$, both the union and intersection operation are defined as a linear function as follows:

$$h_{x_{1\vee 2}} = f_u(h(x_1), h(x_2)) \equiv W_u[h(x_1), h(x_2)] \tag{3}$$

$$h_{x_{1\wedge 2}} = f_t(h(x_1), h(x_2)) \equiv W_t[h(x_1), h(x_2)] \tag{4}$$

where $[h_1, h_2]$ denotes the concatenation operation, W_u, W_t are two matrices to be learnt for union and intersection respectively, and $h_{x_{1\vee 2}}, h_{x_{1\wedge 2}}$ are the fusion codes with label $l_1 \oplus l_2$ and $l_1 \otimes l_2$ respectively.

Substraction Operation. The situation for the substraction operation can be tricky - it is not a good practice to randomly generating a training pairs for it as in the case of union and intersection, because $l_1 \cap l_2 = \emptyset$ is meaningless for $l_1 - l_1 \cap l_2$. To handle this problem, we cascade the substraction to the union operation and define it as follows:

$$h_{x_{1\vee 2\setminus 2}} = f_s(h_{x_{1\vee 2}}, h(x_2)) \equiv W_s[h_{x_{1\vee 2}}, h(x_2)] \tag{5}$$

where W_s is the parameter matrix for the substraction operator, $h_{x_{1\vee 2\setminus 2}}$ is the subtracted code with multi-label $l_1 \oplus l_2 \ominus l_2$.

The advantages of operating codes are three folds: first, through code operator network, feature sets can be augmented in training, which effectively alleviates the imbalance and sparsity problem of multi-label learning; second, one can easily construct useful triplets using codes operation to capture more complicate similarity structure of data. For example, one can use the union operation to combine any two concepts (even though they rarely appear together, and hence it is difficult to obtain their training samples) and to generate the corresponding hashing codes; and finally, complex similarity level can be simulated with the three basic operations. As

we will see, this is useful in handle the multilevel similarity problem in the multi-label context.

3.4 Loss Function

The goal of DSOH is to learn binary codes that preserve multilevel semantic similarity of images. i.e., the codes of *very similar* items should be as close as possible, while the codes of *normally similar* images should be a certain distance away, and those of *dissimilar* images very far away. Besides, to facilitate the code operation at the semantic level, the generated hashing codes should be discriminative. For these purposes, the loss function in this work is designed based on the following criterions: 1) the distance between the learned codes should be adaptive according to their similarity levels; 2) the hashing codes corresponding to different concepts should be distinguishable in the semantic space.

Multi-level similarity loss. One commonly used similarity measurement for hashing codes is the Hamming distance (denoted as d_H), which counts how many bits are different for a given pair of hashing codes b_1 and b_2. Intuitively, the more similar their semantics are (i.e., share more number of labels), the closer their learned codes are. We formalize this as a Lipschitz-like requirement that essentially connects the similarity in the feature space and that in the label space in a meaningful way, as follows,

$$d_L(l_1, l_2) \leq d_H(b_1, b_2) \cdot \frac{1}{m} \tag{6}$$

where m is a scale number whose value is determined by the bits length of underlying hashing codes (see Section 4.2 for details), and d_L is the pairwise distance between the corresponding labels (denoted as l_1 and l_2, respectively) of the two points b_1 and b_2, defined as follows,

$$d_L(l_1, l_2) = \frac{\max\{|l_1|, |l_2|\} - |l_1 \cap l_2|}{\max\{|l_1|, |l_2|\}} \tag{7}$$

where $\|\cdot\|$ denotes the L-1 length of a label vector.

We adopt the scheme of triplets to capture the similarity structure between the hashing codes. Specifically, for a given triplet of images $\{I_1, I_2, I_3\}$ and their labels $\{l_1, l_2, l_3\}$, we first obtain their hashing codes $b_1 = sign(h(f_{CNN}(I_1)))$, $b_2 = sign(h(f_{CNN}(I_2)))$, $b_3 = sign(h(f_{CNN}(I_3)))$, and three derived hashing codes through the CON network [1], i.e., $b_4 = f_u(b_1, b_2)$, $b_5 = f_t(b_1, b_2)$, and $b_6 = f_s(b_4, b_2)$. Hence totally we have six hashing codes and we partition them into three triplets, i.e., $\{b_1, b_2, b_3\}$, $\{b_1, b_2, b_4\}$, and $\{b_1, b_2, b_5\}$, each of which reflects some aspect of the similarity structure and corresponds to a separate triplet loss term in the final loss function.

For the triplet $\{b_1, b_2, b_3\}$, we first select one point from them with most tags as the reference point, based on which we evaluate the similarity between the remaining two points and the reference point using Eq. (7). In this way, we can re-order $\{b_1, b_2, b_3\}$ and denote them as $\{b_{*,1}, b_*, b_{*,2}\}$, where b_* is the reference point, and $b_{*,1}$ is the one that is more similar to b_* than the other ($b_{*,2}$). Finally, we formulate the triplet loss needed as follows,

$$\begin{aligned} L_{tr1}(b_1, b_2, b_3) =& [d_H(b_*, b_{*,1}) - d_H(b_*, b_{*,2}) + \alpha_1]_+ \\ & s.t.\ b \in \{-1, 1\}^k \end{aligned} \tag{8}$$

where $[\cdot]_+ = max\{0, \cdot\}$, and α_1 is the margin and is defined as $\alpha_1 = \frac{|l_* \cap l_{*,1}| - |l_* \cap l_{*,2}|}{|l_*|} \cdot m$. Note that the margin α_1 is adaptive according to their similarity levels.

Similarly, for the group $\{b_1, b_2, b_4\}$ and $\{b_1, b_2, b_5\}$ (where $l_4 = l_1 \cup l_2$ and $l_5 = l_1 \cap l_2$), their relative similarity relationship should respectively satisfy:

$$\begin{aligned} &|l_1| > |l_2| \Rightarrow d_H(b_1, b_4) < d_H(b_1, b_2), d_H(b_2, b_5) < d_H(b_1, b_2) \\ &|l_1| < |l_2| \Rightarrow d_H(b_2, b_4) > d_H(b_1, b_2), d_H(b_1, b_5) < d_H(b_1, b_2) \\ &|l_1| = |l_2| \Rightarrow d_H(b_1, b_4) = d_H(b_2, b_4), d_H(b_1, b_5) = d_H(b_2, b_5) \end{aligned} \tag{9}$$

It can be easily verified that these constraints meet the Lipschitz condition given by Eq. (6). Based on these, the second part of our loss function according to the above two triplets is defined as follows:

$$\begin{aligned} L_{tr2}(b_1, b_2, b_3) = & [y d_H(b_1, b_4) + (1-y) d_H(b_2, b_4) \\ & - d_H(b_1, b_2) + \alpha_2]_+ + [y d_H(b_2, b_5) \\ & + (1-y) d_H(b_1, b_5) - d_H(b_1, b_2) + \alpha_3]_+ \\ & s.t.\ b \in \{-1, 1\}^k \end{aligned} \tag{10}$$

where margin $\alpha_2 = \frac{(yn_1 + (1-y)n_2)^2 - n_3 n_4}{n_3(yn_1 + (1-y)n_2)} \cdot m$ and $\alpha_3 = \frac{|n_1 - n_2| n_4}{n_1 n_2} \cdot m$, $n_1 = |l_1|$, $n_2 = |l_2|$, $n_3 = |l_1 \cup l_2|$, $n_4 = |l_1 \cap l_2|$, and if $n_1 > n_2$ then $y = 1$ otherwise $y = 0$.

Combining Eq. 8 and Eq. 10, we obtain the margin-adaptive triplet loss for the three groups of hashing codes as follows:

$$L_{tr}(b_1, b_2, b_3) = L_{tr1}(b_1, b_2, b_3) + L_{tr2}(b_1, b_2, b_3) \tag{11}$$

Weighted classification loss. We also adopt the following weighted cross-entropy loss to ensure that each individual hashing codes consistent with its own semantic concept,

$$L_{cl}(b, l) = -\sum_{j=1}^{C} \left(w_{pos} \cdot l_j \log(\hat{l_j}) + (1 - l_j) \log(1 - \hat{l_j}) \right) \tag{12}$$

where $\hat{l}$ is the predicted value of DSOH, and w_{pos} is the weight of positive samples.

Finally, the overall loss function for N training triplets $\{b_{i1}, b_{i2}, b_{i3}\}_{i=1}^N$ is defined as follows:

$$\begin{aligned} \mathcal{L} = \sum_{i=1}^{N} \big(& \sum_{j=1}^{3} L_{cl}(b_{ij}, l_{ij}) + \lambda_1 (L_{cl}(b_{i4}, l_{i4}) + L_{cl}(b_{i5}, l_{i5}) \\ & + L_{cl}(b_{i6}, l_{i6})) + \lambda_2 L_{tr}(b_{i1}, b_{i2}, b_{i3}) \big) \end{aligned} \tag{13}$$

where λ_1 and λ_2 are parameters that balance the classification loss of operating codes and the triplet loss.

3.5 Optimization

Note that the problem of Eq. (13) is a discrete optimization problem, which is NP-hard to solve. To address this issue we

[1] Note that in principle we can generate more hashing codes using the CON network but for simplicity and w.l.o.g., here only a set involving two binary codes are used.

use two types of relaxation over the loss function. First, *tanh* activation function is used to approximate the binary codes. Second, Hamming distance is substituted with Euclidean distance, i.e. $d(h_1, h_2) = ||h_1 - h_2||_2^2$. The final loss function Eq. (13) is hence rewritten as follows:

$$\begin{aligned}\mathcal{L} = \sum_{i=1}^{N} \{ \sum_{j=1}^{3} L_{cl}(h_{ij}, l_{ij}) + \lambda_1 (L_{cl}(h_{i4}, l_{i4}) + L_{cl}(h_{i5}, l_{i5}) \\ + L_{cl}(h_{i6}, l_{i6})) + \lambda_2 L_{tr}(h_{i1}, h_{i2}, h_{i3}, l_{i1}, l_{i2}, l_{i3}) \\ + \lambda_3 \sum_{j=1}^{3} (|||h_{i,j}| - \mathbf{1}||_1) \}\end{aligned} \tag{14}$$

where $h_* = h(f_{CNN}(I_*))$ for short, $h_{i4} = f_u(h_{i1}, h_{i2})$, $h_{i5} = f_t(h_{i1}, h_{i2})$,$h_{i6} = f_s(h_{i4}, h_{i2})$. $\mathbf{1}$ is a vector with all element one. $||\cdot||_1$ denotes L1-norm and λ_3 controls the weight of quantization. As this new objective is smooth, the optimization for the whole model is performed using stochastic gradient descent based on Adaptive Moment Estimation (Adam).

4 EXPERIMENTS

4.1 Experimental Settings

We validate the proposed DSOH on three multi-label image datasets, MIRFLICKR25K [9], VOC2012 [4] and Microsoft COCO [17]. MIRFLICKR25K [9] consists of 25,000 multi-label images. Each image is associated 24 classes. In our experiments, samples with no tags are discarded, and 20,006 are remaining. VOC2012 [4] consists of 22,531 images in 20 classes. Since the ground truth labels of the test images are not available, only 11,540 images from its training set are used. For the above two datasets, we randomly sample 2,000 images as the test query set, and the remaining images are used for training. Microsoft COCO [17] contains 82,783 training images and 40,504 testing images. Each image is associated with 90 categories. After pruning images with no category information, we generate 82,081 training samples and 4,956 random test samples.

We perform multi-label retrieval with three kinds of retrieval tasks, which are defined as follows.

- **Task 1** . Retrieve relevant images in the training set using the code of a test image.
- **Task 2** . Retrieve relevant images in the training set using a pair of queries with *Union*.
- **Task 3** . Retrieve relevant images in the training set using a pair of queries with *Substract or Intersect*.

We adopt the commonly-used Average Cumulative Gain (ACG) [10], Normalized Discounted Cumulative Gain (NDCG), and weighted mean Average Precision (weighted mAP) as the performance metric.

4.2 Implementation details

Our DSOH is implemented with Tensorflow[2]. The detailed configurations are illustrated in Table 1. During training, the batch size is set to 64, the learning rate to 0.0001, the λ_1 to 0.01, the λ_2 to 0.1, the λ_3 to 10^{-5}, the w_{pos} is set to 20 for MIRFLICKR25k, VOC2012 and 30 for COCO. The margin parameter m is heuristically set to $2k$ to encourage the codes of dissimilar images to differ in no less than a half of bit length. To make the best use of limited computational resources, all image triplets are generated online from the shuffled training set after each epoch.

[2]https://www.tensorflow.org

Table 1: The configuration of the Deep Semantic Operation Hashing network in experiments.

sub-DHN	
CNN	ResNet-152→2,048 / VGG16→4,096
full1	1,024, tanh
full2	hash code length k, tanh
sub-CON	
Concat.	[k,k]
full3	hash code length k

Table 2: Comparison at Top 100 of End-to-End and Off-the-shelf configuration of DSOH on VOC2012.

Method	NDCG	ACG	mAP$_w$	Train Time(s)
DSOH$_{\text{off-the-shelf}}$	0.7712	1.053	1.073	6.2×10^2
DSOH$_{\text{end2end}}$	0.7877	1.080	1.098	2.8×10^4

Because the CNN model in DSOH is optionally connected, we firstly investigate the influence of joint learning of CNN. We perform CNN fine-tuning and DSOH learning in an end-to-end architecture, named DSOH$_{\text{end2end}}$. Then it was compared with DSOH$_{\text{off-the-shelf}}$ which adopts off-the-shelf CNN feature. CNN models of them are the same VGG16 pre-trained in ImageNet. And two models are trained with 50 epochs. The results on retrieval task 1 are shown in Table 2. We can see that DSOH$_{\text{off-the-shelf}}$ achieves competitive performance to DSOH$_{\text{end2end}}$. While its computation cost is lower than DSOH$_{\text{end2end}}$ by two orders of magnitude. Therefore, considering the time cost, DSOH adopts the off-the-shelf CNN feature in the following experiments.

4.3 Experimental Result

Comparative methods: We compared DSOH with unsupervised hashing LSH [5], SH [27], ITQ [6] and state-of-the-art deep methods MSDH [21], DRSH [31], DMSSPH [28] and our DSOH-NO which removes CON. For a fair comparison, all methods are based on the off-the-shelf 2,048-D ResNet-152 [8] features which are pre-trained on ImageNet with Caffe [11]. LSH, SH, and ITQ were implemented using source codes provided by the authors. To eliminate the effect of the network structures, we implement DMSH, DRSH, DMSSPH and our DSOH with the same network structure as Table 1 shows. All parameters were set optimally based on experimental verification. For retrieval task 2, the fusion code of DSOH-L, DSOH-NO-L and other methods and are empirically obtained by logical OR operation of two query codes, which is consistent with the fusion manner of query label.

Table 3: NDCG@100, ACG@100 and mAP$_w$ of `Task 1` on three datasets with different bits length.

Methods	MIRFLICKR25K				VOC2012				COCO			
	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
	NDCG@100											
DSOH	**0.5307**	**0.5603**	**0.5872**	**0.5966**	**0.7819**	**0.8047**	**0.8195**	**0.8242**	**0.4663**	**0.5178**	**0.5400**	**0.5439**
DSOH-NO	0.4925	0.5359	0.5658	0.5712	0.7664	0.7908	0.8074	0.8077	0.4234	0.4747	0.5101	0.5221
DMSSPH [28]	0.4835	0.4877	0.5004	0.5241	0.7037	0.7136	0.7310	0.7353	0.4337	0.4817	0.5053	0.5026
DRSH [31]	0.4396	0.4774	0.5113	0.5301	0.6895	0.7290	0.7572	0.7785	0.2994	0.3747	0.4267	0.4553
MSDH [21]	0.3950	0.4265	0.4527	0.4602	0.6479	0.7032	0.7081	0.6922	0.3338	0.4066	0.4464	0.4721
ITQ [6]	0.3577	0.4063	0.4297	0.4482	0.5794	0.6418	0.6694	0.6882	0.2761	0.3799	0.4345	0.4674
SH [27]	0.3238	0.3636	0.3757	0.3849	0.5315	0.5689	0.5662	0.5512	0.2809	0.3686	0.4044	0.4244
LSH [5]	0.2377	0.2635	0.3207	0.3723	0.2586	0.3306	0.4512	0.5450	0.1683	0.2370	0.3231	0.4009
	ACG@100											
DSOH	**2.426**	**2.529**	**2.617**	**2.645**	**1.065**	**1.095**	**1.106**	**1.125**	**1.439**	**1.570**	**1.628**	**1.640**
DSOH-NO	2.306	2.454	2.548	2.569	1.040	1.073	1.097	1.095	1.307	1.440	1.544	1.589
DMSSPH [28]	2.225	2.238	2.292	2.318	0.9525	0.9688	0.9875	0.9943	1.355	1.453	1.509	1.523
DRSH [31]	2.200	2.316	2.434	2.475	0.9448	0.9965	1.033	1.064	1.079	1.248	1.362	1.432
MSDH [21]	1.983	2.086	2.166	2.184	0.8733	0.9494	0.9557	0.9368	1.072	1.258	1.367	1.432
ITQ [6]	1.857	2.016	2.081	2.150	0.7959	0.8738	0.9130	0.9362	0.9962	1.222	1.348	1.428
SH [27]	1.681	1.805	1.831	1.875	0.7303	0.7672	0.7509	0.7239	0.9920	1,164	1.233	1.278
LSH [5]	1.346	1.460	1.681	1.873	0.3885	0.4763	0.6340	0.7536	0.6678	0.8434	1.074	1.266
	mAP$_w$@100											
DSOH	**2.467**	**2.566**	**2.656**	**2.683**	**1.87**	**1.112**	**1.123**	**1.142**	**1.464**	**1.599**	**1.657**	**1.671**
DSOH-NO	2.351	2.496	2.589	2.609	1.059	1.094	1.116	1.114	1.344	1.481	1.584	1.629
DMSSPH [28]	2.267	2.294	2.344	2.371	0.9729	0.9921	1.016	1.023	1.376	1.476	1.533	1.551
DRSH [31]	2.216	2.334	2.453	2.504	0.9682	1.023	1.055	1.087	1.085	1.265	1.384	1.462
MSDH [21]	2.018	2.130	2.222	2.256	0.8936	0.9742	0.9862	0.9701	1.091	1.292	1.402	1.472
ITQ [6]	1.886	2.067	2.142	2.213	0.8249	0.9081	0.9527	0.9792	1.024	1.260	1.385	1.470
SH [27]	1.767	1.902	1.945	2.001	0.7917	0.8432	0.8484	0.8398	1.035	1.216	1.295	1.347
LSH [5]	1.393	1.531	1.767	1.978	0.4442	0.5496	0.7139	0.8357	0.7111	0.9015	1.140	1.328

Similarity r	4	3	2	1	0
						
mean d_H	1.372	3.409	4.519	6.194	9.113

Figure 2: The consistency of Hamming Distance d_H and Semantic Similarity r on VOC2012 with 16 bits.

Results on Task 1: From Table 3, we can observe that 1) the proposed DSOH substantially outperforms other compared methods with three evaluation criteria on all used datasets. E.g., The NDCG@100 results of DSOH indicate a 3.2% $\sim$ 4.1% relative increase over the second best baseline on COCO. 2)DSOH outperforms DSOH-NO 3% NDCG scores in average, which means the effectiveness of the code operation. 3) Compared with multilevel similarity-based DMSSPH, DSOH can exploit more similarity information in triplets and discriminative supervised information to capture the multilevel similarity structure accurately.

Besides, it is worth noting that both DSOH and DRSH learn hashing functions based on triplet ranking loss, the results in Table 3 shows that DSOH gains all-around advantages over DRSH on three datasets. To be more specific, the NDCG@100, ACG@100, and weighted mAP@100 results indicate the relative increase of 8.8%$\sim$16.6%, 20.8%$\sim$36% and 20.9%$\sim$37.9% respectively over DRSH on COCO. This effectively demonstrates the benefit of using our adaptive margin loss.

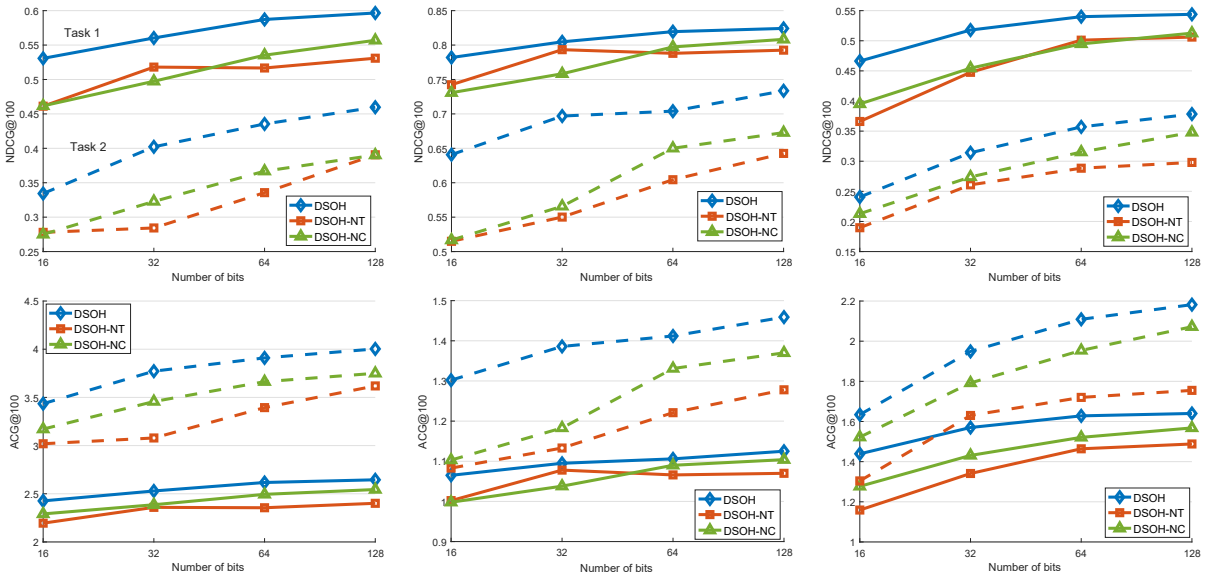
To verify the consistency of hamming distance of learned codes and multilevel semantic similarity, we select all training samples with four tags as reference images and average the hamming distance between them and others samples according to their semantic similarity. Fig. 2 shows the results. we can see the d_H exactly satisfies the requirement of Eq. (6).

Results on Task 2: From table 4, we can observe that: 1) DSOH leads to superior results with 16.3%,17.9% and 10.9% improvements (NDCG@100) over the best-performing comparison methods on the three datasets with 128 bits. It demonstrates that our hashing code is operable to perform complex retrieval task. 2) Because of the non-considering on the consistency between the logical operation of hashing codes and corresponding semantic label, simple OR operation of hashing code from DSOH-L cannot accomplish complex semantic retrieval task well. Besides, DSOH-NO-L is inferior to DSOH-L because of no CON. 3) With code length increasing, the performance of our DSOH intends to be stable, which implies that a proper length of code can obtain optimal performance while longer is useless.

Furthermore, compared with deep baselines DRSH and DMSSPH, we can find that 1) since DMSSPH considering p-reserving multilevel semantic similarity with certain distance

Table 4: NDCG@100, ACG@100 and mAP$_w$ of Task 2 on three datasets with different bits length.

Methods	MIRFLICKR25K				VOC2012				Microsoft COCO			
	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
	NDCG@100											
DSOH	**0.3343**	**0.4022**	**0.4354**	**0.4597**	**0.6408**	**0.6969**	**0.7040**	**0.7336**	**0.2406**	**0.3141**	**0.3570**	**0.3783**
DSOH-L	0.1603	0.2093	0.2578	0.2994	0.4194	0.4518	0.6202	0.6130	0.2075	0.2282	0.2743	0.2584
DSOH-NO-L	0.1386	0.1933	0.2045	0.2538	0.4339	0.4962	0.5381	0.5579	0.1563	0.1857	0.1785	0.2521
DMSSPH [28]	0.1986	0.2347	0.2716	0.2184	0.3994	0.4778	0.5061	0.4682	0.1379	0.2332	0.2389	0.2687
DRSH [31]	0.1729	0.1935	0.2382	0.2965	0.3612	0.4430	0.4844	0.5539	0.1274	0.1635	0.1974	0.2306
MSDH [21]	0.1461	0.1519	0.1685	0.2094	0.3695	0.3818	0.3654	0.4149	0.1509	0.1742	0.2098	0.2057
	ACG@100											
DSOH	**3.297**	**3.724**	**3.905**	**3.939**	**1.302**	**1.386**	**1.412**	**1.459**	**1.634**	**1.949**	**2.109**	**2.182**
DSOH-L	2.288	2.651	3.056	3.268	0.8833	0.9708	1.264	1.250	1.452	1.497	1.676	1.655
DSOH-NO-L	2.048	2.628	2.691	3.087	0.9228	1.033	1.102	1.152	1.131	1.285	1.214	1.599
DMSSPH [28]	2.455	2.795	2.976	2.562	0.8666	1.002	1.041	0.9911	1.005	1.531	1.478	1.634
DRSH [31]	2.464	2.659	2.971	3.316	0.7803	0.9658	1.032	1.158	1.020	1.246	1.347	1.499
MSDH [21]	2.151	2.137	2.307	2.646	0.7916	0.8598	0.8128	0.9240	1.091	1.194	1.396	1.334
	mAP$_w$@100											
DSOH	**3.499**	**3.846**	**3.991**	**4.087**	**1.353**	**1.445**	**1.457**	**1.506**	**1.640**	**1.970**	**2.142**	**2.217**
DSOH-L	2.322	2.684	3.071	3.311	0.9151	0.9850	1.301	1.284	1.446	1.509	1.694	1.670
DSOH-NO-L	2.113	2.632	2.711	3.113	0.9606	1.059	1.144	1.186	1.149	1.317	1.223	1.619
DMSSPH [28]	2.501	2.795	3.072	2.633	0.9252	1.071	1.104	1.038	1.012	1.545	1.478	1.649
DRSH [31]	2.424	2.640	2.951	3.318	0.7946	0.9720	1.053	1.185	1.009	1.217	1.348	1.497
MSDH [21]	2.147	2.155	2.327	2.675	0.8091	0.8769	0.8389	0.9389	1.102	1.201	1.401	1.339

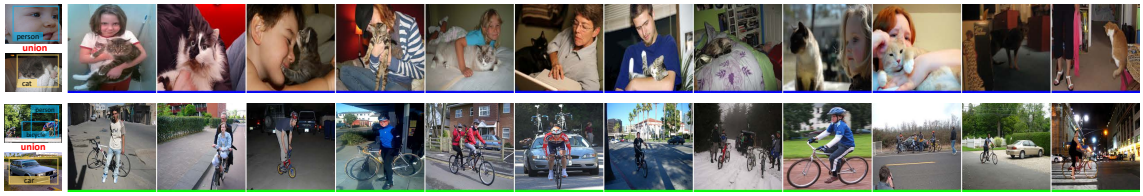


(a) MIRFLICKR25K (b) VOC2012 (c) Microsoft COCO

Figure 3: NDG@100 and ACG@100 of two retrieval tasks with different components and various hashing bits.

constraint in Hamming space, it produces a superior performance to DRSH. 2) Although both DSOH and DMSSPH use adaptive margin in loss function to capture multilevel similarity, DSOH leads to higher accuracies. This performance gap between them may be caused by 1) the exploitation of discriminative supervised information for fused code in DSOH can guide more effective fusion, 2) comprehensive similarity relationship between original codes and fused code in triplet is helpful to improve the generalization of DSOH, 3) the operation of code in DSOH is actually a linear transformation, which can select relevant components from pre-operated codes to integrate and avoid introducing noise in fusion process, while logical OR cannot.

Some retrieval results of Task 2 and Task 3 are presented in Figure 4. From sub-figure (a), we can see the semantic concepts of the query pairs are well fused, and meaningful



(a) Semantic retrieval with hashing codes union operation in DSOH

(b) Semantic retrieval with hashing codes subtract operation in DSOH

(c) Semantic retrieval with hashing codes intersect operation in DSOH

Figure 4: Retrieval examples on VOC2012 with 128 bits. Top 12 results for each query pairs are shown. Results with red border denotes $|l_q \cap l_{res}| = 1$, blue denotes $|l_q \cap l_{res}| = 2$ and green denotes $|l_q \cap l_{res}| > 2$.

results are returned. For instance, 'car + motorbike' returns 'A motorbike parked beside a car.' It is worth noted that DSOH could handle more complex semantic fusion. E.g., 'Person is riding a bicycle + car' returns 'a person rides a bicycle in front of a car.' Besides, from sub-figure (b), it can be seen the semantic concepts of the first query hashing code is partially removed by 'subtracting' the second query code. E.g., 'car, person, dog - car' returns 'A person is playing with a dog.' While in sub-figure (c), the overlap semantic concepts are used for more precise retrieval.

4.4 Effect of Model Configuration

To analyze the effectiveness of different loss term in the proposed DSOH, we separately remove L_{cl} and L_{tr} with other parameters remained to evaluate their influence on the final performance. These two models are called DSOH-NC and DSOH-NT. Here NC denotes no classification loss L_{cl} term and NT denotes no multi-level triplet loss L_{tr} term. Figure 3 shows the result of two tasks.

We can see that jointly using the margin-adaptive triplet loss and classification loss can apparently improve the ranking quality of top-100 relevant items in terms of NDCG, ACG and mAP at the expense of the averaged ranking performance, as it adaptively assigns margins according to the related multilevel similarity and simultaneously encodes rich discriminative information. Besides, for Task 1, with the increasing of the bits number, the performance of DSOH-NC increases firstly and then tends to be flat, which indicates the term L_{cl} cannot take advantage of longer bits to boost performance. For Task 2, DSOH-NT performances worse than DSOH-NC which implies that the term L_{tr} is more important than L_{cl} to capture multilevel similarity.

5 CONCLUSIONS

In this paper, we have presented a novel deep hashing method, named DSOH, to learn multilevel similarity-preserving and operable binary codes for multi-label image retrieval. Unlike many previous works that ignore the *intention gap* issue, we propose a new Code Operation Network to perform the 'union,' 'intersect' and 'subtract' operations on semantic concepts of multiple query image codes, which effectively enables users to expand their query intention in a more flexible and friendly manner. The whole system is learned in an end-to-end way with the help of a newly designed adaptive-margin based triplet loss function. Our experimental results demonstrate that the proposed method is able to learn useful multilevel semantic similarity-preserving binary codes and achieves state-of-the-art retrieval performance on three popular datasets.

ACKNOWLEDGMENTS

This work is partially supported by National Natural Science Foundation of China (61672280,61373060,61732006), the Pre-research fund of Equipments of China, Jiangsu 333 Project (BRA2017377) and Qing Lan Project.

REFERENCES

[1] Jiale Bai, Bingbing Ni, Minsi Wang, Yang Shen, Hanjiang Lai, Chongyang Zhang, Lin Mei, Chuanping Hu, and Chen Yao. 2017. Deep Progressive Hashing for Image Retrieval. In *Proceedings of the 2017 ACM on Multimedia Conference, MM 2017, Mountain View, CA, USA, October 23-27, 2017*. 208–216.

[2] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Philip S. Yu. 2017. HashNet: Deep Learning to Hash by Continuation. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. 5609–5618.

[3] Yueqi Duan, Jiwen Lu, Ziwei Wang, Jianjiang Feng, and Jie Zhou. 2017. Learning Deep Binary Descriptor with Multi-quantization. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 4857–4866.

[4] Mark Everingham, Luc Gool, Christopher K Williams, John Winn, and Andrew Zisserman. 2010. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision* 88, 2 (2010), 303–338.

[5] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. 1999. Similarity Search in High Dimensions via Hashing. In *International Conference on Very Large Data Bases*. 518–529.

[6] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. 2013. Iterative Quantization: A Procrustean Approach to Learning Binary Codes for Large-Scale Image Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 12 (2013), 2916–2929.

[7] Albert Gordo and Diane Larlus. 2017. Beyond Instance-Level Image Retrieval: Leveraging Captions to Learn a Global Visual Representation for Semantic Retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*. 5272–5281.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. 770–778.

[9] Mark J. Huiskes and Michael S. Lew. 2008. The MIR flickr retrieval evaluation. In *ACM International Conference on Multimedia Information Retrieval*. 39–43.

[10] Kalervo Järvelin and Jaana Kekäläinen. 2000. IR evaluation methods for retrieving highly relevant documents. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 41–48.

[11] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross B. Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional Architecture for Fast Feature Embedding. In *Proceedings of the ACM International Conference on Multimedia, MM '14, Orlando, FL, USA, November 03 - 07, 2014*. 675–678.

[12] Gunhee Kim, Seungwhan Moon, and Leonid Sigal. 2015. Ranking and retrieval of image sequences from multiple paragraph queries. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1993–2001.

[13] H. Lai, P. Yan, X. Shu, Y. Wei, and S. Yan. 2016. Instance-Aware Hashing for Multi-Label Image Retrieval. *IEEE Transactions on Image Processing* 25, 6 (2016), 2469–2479.

[14] Qi Li, Zhenan Sun, Ran He, and Tieniu Tan. 2017. Deep Supervised Discrete Hashing. In *International Conference on Neural Information Processing Systems*.

[15] Wu-Jun Li, Sheng Wang, and Wang-Cheng Kang. 2016. Feature Learning Based Deep Supervised Hashing with Pairwise Labels. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*. 1711–1717.

[16] Jie Lin, Olivier Morère, Antoine Veillard, Ling-Yu Duan, Hanlin Goh, and Vijay Chandrasekhar. 2017. DeepHash for Image Instance Retrieval: Getting Regularization, Depth and Fine-Tuning Right. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval, ICMR 2017, Bucharest, Romania, June 6-9, 2017*. 133–141.

[17] Tsung Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollr, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision*. 740–755.

[18] Haomiao Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. 2016. Deep Supervised Hashing for Fast Image Retrieval. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[19] Li Liu, Ling Shao, Fumin Shen, and Mengyang Yu. 2017. Discretely Coding Semantic Rank Orders for Supervised Image Hashing. In *IEEE Conference on Computer Vision and Pattern Recognition*. 5140–5149.

[20] Li Liu, Fumin Shen, Yuming Shen, Xianglong Liu, and Ling Shao. 2017. Deep Sketch Hashing: Fast Free-hand Sketch-Based Image Retrieval. In *Computer Vision and Pattern Recognition*.

[21] Jiwen Lu, Venice Erin Liong, and Jie Zhou. 2017. Deep Hashing for Scalable Image Search. *IEEE Transactions on Image Processing* 26, 5 (2017), 2352–2367.

[22] Yadong Mu and Zhu Liu. 2017. Deep Hashing: A Joint Approach for Image Signature Learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*. 2380–2386.

[23] Amaia Salvador, Nicholas Hynes, Yusuf Aytar, Javier Marín, Ferda Ofli, Ingmar Weber, and Antonio Torralba. 2017. Learning Cross-Modal Embeddings for Cooking Recipes and Food Images. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 3068–3076.

[24] Omar Seddati, Stéphane Dupont, and Saïd Mahmoudi. 2017. Quadruplet Networks for Sketch-Based Image Retrieval. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval, ICMR 2017, Bucharest, Romania, June 6-9, 2017*. 184–191.

[25] Ge Song and Xiaoyang Tan. 2017. Hierarchical deep hashing for image retrieval. *Frontiers Comput. Sci.* 11, 2 (2017), 253–265.

[26] Giorgos Tolias and Ondrej Chum. 2017. Asymmetric Feature Maps with Application to Sketch Based Retrieval. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. 6185–6193.

[27] Yair Weiss, Antonio Torralba, and Rob Fergus. 2008. Spectral hashing. In *International Conference on Neural Information Processing Systems*. 1753–1760.

[28] Dayan Wu, Zheng Lin, Bo Li, Mingzhen Ye, and Weiping Wang. 2017. Deep Supervised Hashing for Multi-Label and Large-Scale Image Retrieval. In *ACM International Conference on Multimedia Retrieval*. 150–158.

[29] Changcheng Xiao, Changhu Wang, Liqing Zhang, and Lei Zhang. 2015. Sketch-based Image Retrieval via Shape Words. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, Shanghai, China, June 23-26, 2015*. 571–574.

[30] Bo Zhao, Jiashi Feng, Xiao Wu, and Shuicheng Yan. 2017. Memory-Augmented Attribute Manipulation Networks for Interactive Fashion Search. In *IEEE Conference on Computer Vision and Pattern Recognition*. 6156–6164.

[31] Fang Zhao, Y. Huang, L. Wang, and Tieniu Tan. 2015. Deep semantic ranking based hashing for multi-label image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1556–1564.

[32] Wengang Zhou, Houqiang Li, and Qi Tian. 2017. Recent Advance in Content-based Image Retrieval: A Literature Survey. *arXiv* (2017).