Computer Vision and Image Understanding 193 (2020) 102916

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Deep code operation network for multi-label image retrieval ☆

Ge Song, Xiaoyang Tan *

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
MIIT Key Laboratory of Pattern Analysis and Machine Intelligence, China
Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 211106, China

ARTICLE INFO

Communicated by Nikos Paragios

Keywords:
Multi-label image retrieval
Hashing
Deep learning

ABSTRACT

Deep hashing methods have been extensively studied for large-scale image search and achieved promising results in recent years. However, there are two major limitations of previous deep hashing methods for multi-label image retrieval: the first one concerns the flexibility for users to express their query intention (so-called the intention gap), and the second one concerns the exploitation of rich similarity structures of the semantic space (so-called the semantic gap). To address these issues, we propose a novel Deep Code Operation Network (CoNet), in which a user is allowed to simultaneously present multiple images instead of a single one as his/her query, and then the system triggers a series of code operators to extract the hidden relations among them. In this way, a set of new queries are automatically constructed to cover users' real complex query intention, without the need of explicitly stating them. The CoNet is trained with a newly proposed margin-adaptive triplet loss function, which effectively encourages the system to incorporate the hierarchical similarity structures of the semantic space into the learning procedure of the code operations. The whole system has an end-to-end differentiable architecture, equipped with an adversarial mechanism to further improve the quality of the final intention representation. Experimental results on four multi-label image datasets demonstrate that our method significantly improves the state-of-the-art in performing complex multi-label retrieval tasks with multiple query images.

1. Introduction

Image retrieval, aiming to search images displaying the same objects categories or visual content as in a query image, is the central technique to organize and access large-scale image databases. This field has gained increasing attention from both academia and industry in recent years due to its wide usage in real-world applications (e.g., product search, automatic image annotation). As for this task, three types of difficulties commonly encountered make it challenging in practice (Zhou et al., 2017). The first is the intrinsic inconsistent of similarity between low-level visual features and high-level semantic concepts, which is known as the semantic gap. The second one is related to the storage and retrieval efficiency problem when dealing with large-scale data. Last but not the least, a user usually suffers difficulty to precisely express his or her expected visual content by a query at hand, which is referred to intention gap in the literature and is less studied compared to the previous items.

Extensive efforts have been devoted to addressing these issues and one of the most popular techniques is the method of deep hashing (Wu et al., 2017; Lu et al., 2017; Lai et al., 2016; Liu et al., 2016; Lin et al., 2017; Cao et al., 2017; Mu and Liu, 2017; Bai et al., 2017; Duan et al., 2017; Li et al., 2016; Tang et al., 2018; Song et al., 2018b). Deep hashing maps images into compact similarity-preserved binary codes by combining the advantages of the feature learning capability of deep convolutional neural networks and the low storage and computational cost of the hashing method. Conceptually, the deep hashing method solves both the semantic gap and the computational efficiency issue of image retrieval but leaves 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 describe the complex query intention of a user, especially when the intention contains multiple semantics. In literatures, the intention gap is often tackled with multiple alternative query formations, e.g., sketch-based (Liu et al., 2017b; Seddati et al., 2017; Xiao et al., 2015; Tolias and Chum, 2017) and image–text (Gordo and Larlus, 2017; Kim et al., 2015). Indeed, these alternatives serve to be the complement to the original query, which allows the user to express query intention in more flexible ways. Unfortunately, these methods may not always work, for example, under situations when users are not skilled at sketching. They also

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.cviu.2020.102916.

* Corresponding author.
E-mail addresses: sunge@nuaa.edu.cn (G. Song), x.tan@nuaa.edu.cn (X. Tan).

https://doi.org/10.1016/j.cviu.2020.102916
Received 28 July 2019; Received in revised form 25 January 2020; Accepted 28 January 2020
Available online 4 February 2020

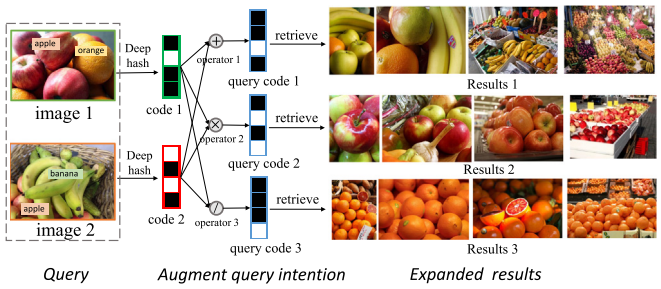


Fig. 1. Our CoNet applies various code operators on multiple query images to capture the hidden relations among them, which helps the users to express their intentions more freely. For example, by *Unioning* 'apples and oranges' and 'apples and bananas', the CoNet essentially constructs a complex query of 'a variety of fruits, including apples, bananas, and oranges'. This can also be seen as a new query expansion mechanism but in a learnable way.

impose an extra burden on learning algorithms to handle different modal data. Besides, nowadays, many images are often associated with more than one tag (referred to multi-label), which provides a valuable rich semantic description of the corresponding image. Previous work (Gordo and Larlus, 2017) has shown that there exists a 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. Query images with multiple labels thus may enlarge the intention gap, as the multiple labels would increase the entropy of the user's intention over those labels.

To alleviate such problems, 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 'apples and oranges' image and 'a basket of apples and bananas' image) to express more complex query intention (e.g., 'a variety of fruits, including apples, bananas, and oranges'). However, directly fusing the low-level features of query images is not easy and may introduce noise, which makes it hard to extract specific query intention of these multiple images. Fortunately, in most deep supervised hashing schemes, since the hash layer commonly lies on the top of the networks, the learned code contains more certain semantic concepts (query intention) than original images. Hence it makes sense to deal with the issue of the intention gap at this level via the operations of hash codes. As illustrated in Fig. 1, this effectively relaxes the burden for users to express their complex query intention explicitly.

Another issue lies in the inherent complex similarity of multi-labels, which involves the measurement of similarity not only at the image-level but at the semantic (label) level as well. Many early deep hashing schemes (Liu et al., 2017b; Li et al., 2017; Liu et al., 2017a, 2016; Song and Tan, 2017) are designed for single-label images, which only concerns about the preservation of simple binary semantic similarity, but the Hamming distance between the learned codes cannot accurately reflect the hierarchical relevance in semantic space. Several recent deep hashing methods (Zhao et al., 2015; Wang et al., 2015; Wu et al., 2017; Lu et al., 2017; Lai et al., 2016) have specifically devoted to the topic of multi-label image retrieval. One popular way of them 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 individual concepts (e.g., the 'person' and 'cat' in the previous examples) less distinguishable in the representation space. Moreover, due to the imbalance and sparsity problem commonly encountered in multi-labels, 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.

To address the above issues, in this work we introduce a novel deep hashing-based method, termed deep Code Operation Network (CoNet). An overview of our approach is illustrated in Fig. 2. Particularly, to ease the burden of the user's expression of query intention, multiple query images are allowed to submit to the system at one time, and then a specific code operation sub-network is invoked to interpret these and to compose new query codes using a set of pre-learned code operators. To facilitate the learning of the desired code operations, we propose a new margin-adaptive triplet loss, which effectively explores and exploits the multi-level similarity structure of the semantic space. The whole system has an end-to-end differentiable architecture, equipped with the adversarial mechanism (Ganin et al., 2016) to further improve the quality of the final intention representation. Compared with existing works, the main contributions of the proposed CoNet method are summarized as follows.

- In contrast to the previous methods that ignore the severe *intention gap* problem in retrieval, the proposed CoNet can explicitly perform the 'union', 'intersect', and 'subtract' fusions on semantic concepts of multiple query image codes, which allows a user to expand their query intention in a more flexible and friendly manner, i.e., uploading multiple query images.
- We connect the similarity in the learned Hamming space and that in the semantic space by a Lipschitz-like requirement. Based on this requirement, we presented the margin-adaptive triplet loss to learn hash functions for multi-level semantic similarity preserved hashing codes.
- We seamlessly integrate the existing adversarial learning framework into the CoNet, such that our method can jointly optimize each part and generate semantically well-fused codes, which is significant for performing a query with multiple images.
- Extensive experiments on several popular multi-label image datasets, including MIRFLICKR25K (Huiskes and Lew, 2008), VOC2012 (Everingham et al., 2010), NUS-WIDE (Chua et al., 2009) and Microsoft COCO (Lin et al., 2014), show that the proposed approach achieves state-of-the-art performance in performing complex retrieval tasks with multiple query images.

The remainder of this paper is organized as follows. We first discuss related work in Section 2. The CoNet model is detailed in Section 3, and experimental results are given in Section 4. We draw our conclusions in Section 5.

2. Related work

In this section, we briefly review related works concerning the intention gap and discuss some of the hashing methods for multi-label image retrieval.

2.1. User intention analysis

In the image retrieval field, there mainly are two ways to reduce the retrieval difficulty of the low-quality query and accurately capture users' search intention. One way is exploiting users' relevance feedback (Tang et al., 2012; Qian et al., 2016; Chen et al., 2018), which includes three types of feedback: explicit feedback, implicit feedback, and blind feedback. Explicit feedback requires users to label multiple positive/negative examples directly and learns a query-specific visual similarity metric from the selected examples to rank images, whereas the implicit feedback (Chen et al., 2018) is inferred from a user's behavior. Undoubtedly, these two feedbacks will increase users' burden and induce inevitable delays (because of the interaction between users and the system). Blind relevance feedback (Qian et al., 2016), also referenced as pseudo-relevance feedback, acquire feedback automatically without interactions with users. It carries out routine retrieval and assumes that some of the top N results are relevant. This method is more appropriate due to its real-time. However, because the blind

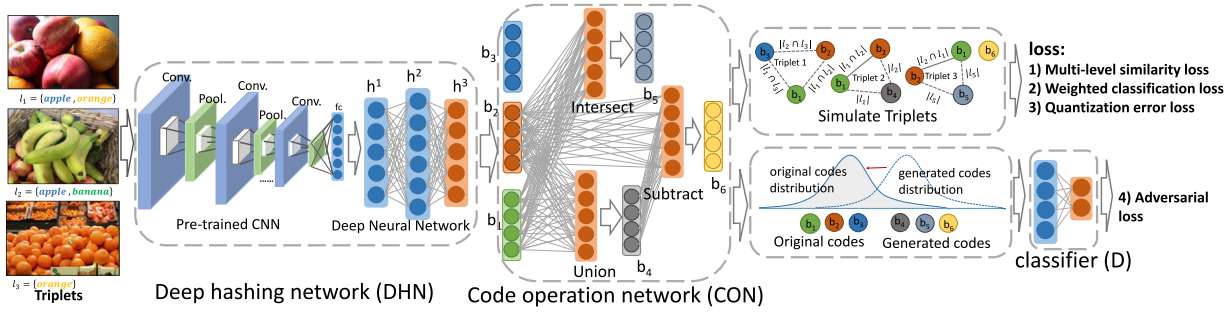


Fig. 2. The architecture of the proposed deep code operation network (CoNet), which includes two sub-networks, one deep hashing network (DHN) and one code operation network (CON). The DHN sub-network takes a triplet images I_1, I_2, I_3 as input and outputs correspondingly three hashing codes b_1, b_2, b_3, which are then processed by the CON sub-network with three types of code operations, i.e., union, subtraction and intersection, to generate b_4, b_5, b_6, respectively. Based on them, three triplets are constructed with the proposed margin-adaptive triplet loss and are used to guide the optimizing procedure. Furthermore, a classifier D is exploited to help the training of the CON sub-network under the adversarial learning framework.

relevance is not dependent on the user, this may lead to unsatisfactory retrieval results in some cases.

Another way is the query expansion (Kim et al., 2015; Gordo and Larlus, 2017; Zhao et al., 2017; Salvador et al., 2017), i.e., providing additional information to make the query more detailed. Since the type of the added information can be different, there are a wide variety of query expansion methods. Kim et al. (2015) noted that taking the form of a single image as the query may be insufficient to reflect user intention in some situations. Therefore they propose a method to rank and retrieve image sequences from a query which consists of multiple images, sentences, or paragraphs. Gordo and Larlus (2017) 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. Zhao et al. (2017) 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. Besides, some technologies are helpful to enhance query quality. The cross-modal searching can directly perform the query of other modalities data via a text description, e.g., the shared predictive deep quantization model (Yang et al., 2018) for text-based image retrieval. The visual caption or analysis (Song et al., 2018c) can extract the semantic description of the query to reduce the uncertainty of query intention, e.g., the multi-modal stochastic RNNs (Song et al., 2018a) for richer captions of visual content. Towards specific image retrieval tasks, more professional information about query need be prepared to capture certain intention for high-quality retrieval, e.g., fine-grained segmentation for very high-resolution remote sensing image retrieval (Chaudhuri et al., 2019). 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.

2.2. Hashing methods for multi-label image retrieval

Hashing method (Gionis et al., 1999; Yuan et al., 2018; Çakir et al., 2017; Cao et al., 2018; Li et al., 2018; Liong et al., 2018; Wang et al., 2019; Ma et al., 2019; Luo et al., 2018) seeks to encode high-dimensional features into compact binary codes, hence enabling high-speed similarity search with Hamming distances. Recently, several hash schemes (Zhao et al., 2015; Wang et al., 2015; Wu et al., 2017; Lu et al., 2017; Lai et al., 2016; Zhang et al., 2018) have been proposed for multi-label image retrieval. To consider multi-labels, Wang et al. (2015) proposed the MLSH (Multi-label Least-Squares Hashing) method, which utilizes the equivalent form of CCA and least-squares to project original multi-label data into lower-dimensional space and learns the project matrix in this space to get final binary codes of data. However, the MLSH is not an end-to-end model, which hence limits its performance. The DSRH (Deep Semantic Ranking based Hashing) (Zhao et al., 2015) 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 in prevalent deep networks. IAH (Instance-Aware Hashing) (Lai et al., 2016) focuses on learning instance-aware hashing for multi-label images with weighted triplet loss functions, in which images are represented by multiple pieces of binary semantic codes corresponds to different categories. Similar to IAH, Zhang et al. (2018) propose ISDH (Instance Similarity Deep Hashing), they define the pairwise similarity into an instance similarity which is quantified into a percentage based on the normalized semantic labels, and learn hash functions with weighted cross-entropy loss and a minimum mean square error loss. MSDH (Multi-label Supervised Deep Hashing) (Lu et al., 2017) uses multi-layer non-linear transformations as the hashing function and learns similarity via weighted pairwise loss, where the weight is computed by the similarity of labels to reflect the different levels, this idea is further expanded to cross-modal retrieval field (Liong et al., 2018). SLDH (Ma et al., 2019) also models hash functions as a multi-layer neural network and optimizes it by minimizing the weighted pairwise loss. While Wu et al. (2017) present DMSSPH (Deep Multilevel Semantic Similarity Preserving Hashing) to learn multilevel similarity preserving codes, in which they design a pairwise loss with adaptive margins that imposes constraints on the distance between the learned codes. Unlike above methods, based on the observing that ranking with the discrete Hamming distance naturally results in ties, He et al. (2017) propose to use a tie-aware version of multi-label ranking metric NDCG (Normalized Discounted Cumulative Gain) in the learning of supervised hashing and directly optimize it with stochastic gradient ascent. Although most of the aforementioned supervised hashing methods are successful in performing multi-label semantic retrieval to some extent, seldom of them have fully kept the multi-level similarity of multi-labels and attempted to exploit the learned codes to express complex query intention for mitigating the inherent intention gap of retrieval task.

3. The proposed approach

In this section, we first give the notations used in this paper. Then the configuration of the proposed CoNet is detailed.

3.1. 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 also represented by a D-dimensional visual features $x_i \in \mathcal{R}^D$. 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 multi-level semantic similarity among images is preserved. Besides, to express more complex queries, we define three types of basic

code operators, i.e., union, subtraction and intersection (more complex operators can be composed of these simple operators) to manipulate codes at the semantic level., 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 $f_\oplus$ fuses them into one $h_\oplus = f_\oplus(h(x_1), h(x_2))$ with label $l_\oplus = l_1 \oplus l_2 = l_1 \cup l_2$; a intersect $f_\otimes$ to fuse them into codes $h_\otimes = f_\otimes(h(x_1), h(x_2))$ with label $l_\otimes = l_1 \otimes l_2 = l_1 \cap l_2$; and a subtractor $f_\ominus$ subtracts $h(x_1)$ with $h(x_2)$ to give $h_\ominus = f_\ominus(h(x_1), h(x_2))$ with label $l_\ominus = l_1 \ominus l_2$, which is equal to $l_1 - l_1 \cap l_2$ if $|l_1| > |l_1 \cap l_2|$, and l_1 otherwise.

3.2. Deep code operation network

Our deep code operation network (CoNet) includes two sub-networks (see Fig. 2), deep hashing network for hash learning, and code operation network for manipulating hash codes.

3.2.1. Deep hashing network

Deep hashing network (DHN) consists of a CNN model and a deep neural network (DNN) with three fully-connect layers. Let x denotes feature vectors from the outputs of the last layer of CNN. W_m is the weight of the mth layer in DNN. The hash codes b of image I can be obtained as the following process:

$$\begin{aligned} x &= CNN(I) \\ h(x) &= \tanh\big(W_2 \cdot \tanh(W_1 x + bias)\big) \end{aligned} \tag{1}$$

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

We adopted the commonly-used VGG or ResNet pre-trained on the ImageNet dataset as the CNN model. Notably, the connection of this CNN model in CoNet is optional, which has no significant influence on the final performance in experiments.

3.2.2. Code operation network

The code operation network (CON) implements the defined three basic code operators based on the outputs of the above deep hashing network, which makes the codes are dependent and can be used collaboratively to express more complex queries.

Implementation of code operator. Given two hashing codes $h(x_1)$, $h(x_2)$, all of the union, intersection, subtraction operators are implemented as linear functions as follows:

$$h_{x_{1\oplus 2}} = f_\oplus(h(x_1), h(x_2)) \equiv W_\oplus[h(x_1), h(x_2)] \tag{3}$$

$$h_{x_{1\otimes 2}} = f_\otimes(h(x_1), h(x_2)) \equiv W_\otimes[h(x_1), h(x_2)] \tag{4}$$

$$h_{x_{1\ominus 2}} = f_\ominus(h(x_1), h(x_2)) \equiv W_\ominus[h(x_1), h(x_2)] \tag{5}$$

where $[h(x_1), h(x_2)]$ denotes the concatenation operation, $W_\oplus$, $W_\otimes$, $W_\ominus \in \mathcal{R}^{K\times 2K}$ are three parameter matrices to be learnt for union, intersection, subtraction respectively, and $h_{x_{1\oplus 2}}$, $h_{x_{1\otimes 2}}$, $h_{x_{1\ominus 2}}$ are the fusion codes with label $l_1 \oplus l_2$, $l_1 \otimes l_2$, $l_1 \ominus l_2$ respectively.

Notably, the situation for the subtraction operation can be tricky — it is not a good practice to randomly generate 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 subtraction to the union operation as follows:

$$h_{x_{1\oplus 2\ominus 2}} = f_s(h_{x_{1\oplus 2}}, h(x_2)) \equiv W_\ominus[h_{x_{1\oplus 2}}, h(x_2)] \tag{6}$$

where $h_{x_{1\oplus 2\ominus 2}}$ 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 to handle the multilevel similarity problem in the multi-label context.

3.3. Loss function

To learn semantic operable hash codes, we design the loss function based on the following criterions. (1) **Multilevel similarity preserving.** The learned 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. i.e., the distance between the learned codes should be adaptive according to their semantic similarity levels. (2) **Semantic discriminative.** For the facilitation of the code operation at the semantic level, the hashing codes should be distinguishable in the semantic space. (3) **Consistent distribution.** Since the outputs of the code operator need to retrieve their nearest neighbors in the input space of code operator, they should not be distinguished unless in semantic level, i.e., the inputs and outputs of the code operation network should have the same distribution. According to the above criterions, we introduce three losses.

3.3.1. 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. Based on the multilevel similarity preserving criterion, the more similar their semantics are (i.e., share more labels), the closer their learned codes are, or a small change of semantic similarity should reflect a slight change in Hamming distance. Therefore, we formalize this as a Lipschitz-like requirement that essentially connects the similarity in the learned Hamming space and that in the label space in a meaningful way, as follows,

$$d_L(l_1, l_2) \le d_H(b_1, b_2) \cdot \frac{1}{m} \tag{7}$$

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,

$$d_L(l_1, l_2) = \frac{\max\{|l_1|, |l_2|\} - |l_1 \cap l_2|}{\max\{|l_1|, |l_2|\}} \tag{8}$$

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 $b_1 = sign(h(CNN(I_1)))$, $b_2 = sign(h(CNN(I_2)))$, $b_3 = sign(h(CNN(I_3)))$, and three derived hashing codes through the CON network,[1] i.e., $b_4 = f_\oplus(b_1, b_2)$, $b_5 = f_\otimes(b_1, b_2)$, and $b_6 = f_\ominus(b_4, b_2)$. 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.

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. (8). 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

[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.

that is more similar to b_* than the other $b_{*,2}$. Finally, we formulate the margin-adaptive triplet loss needed as follows,

$$L_{tr1}(b_1, b_2, b_3) = [d_H(b_*, b_{*,1}) - d_H(b_*, b_{*,2}) + \alpha_1]_+ \quad s.t.\ b \in \{-1, 1\}^k \tag{9}$$

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:

$$\begin{aligned} &|l_1| > |l_2| \Rightarrow d_H(b_1, b_4) < d_H(b_1, b_2), d_H(b_2, b_5) < d_H(b_1, b_2) \\ &|l_1| < |l_2| \Rightarrow d_H(b_2, b_4) > d_H(b_1, b_2), d_H(b_1, b_5) < d_H(b_1, b_2) \\ &|l_1| = |l_2| \Rightarrow d_H(b_1, b_4) = d_H(b_2, b_4), d_H(b_1, b_5) = d_H(b_2, b_5) \end{aligned} \tag{10}$$

It can be easily verified that these constraints meet the Lipschitz condition given by Eq. (7). Based on these, the second part of multi-level similarity loss according to the above two triplets is defined as follows:

$$\begin{aligned} L_{tr2}(b_1, b_2, b_3) = &\big[y d_H(b_1, b_4) + (1-y) d_H(b_2, b_4) \\ &- d_H(b_1, b_2) + \alpha_2\big]_+ + [y d_H(b_2, b_5) \\ &+ (1-y) d_H(b_1, b_5) - d_H(b_1, b_2) + \alpha_3]_+ \\ &s.t.\ b \in \{-1, 1\}^k \end{aligned} \tag{11}$$

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 Eqs. (9) and (11), we obtain the margin-adaptive triplet loss for the three groups of hashing codes as follows:

$$L_{tr}(b_1, b_2, b_3) = L_{tr1}(b_1, b_2, b_3) + L_{tr2}(b_1, b_2, b_3) \tag{12}$$

3.3.2. Weighted classification loss

Based on the semantic discriminative criterion, we adopt the following weighted cross-entropy loss to ensure that each hashing codes consistent with its semantic concept,

$$L_{cl}(b, l) = -\sum_{j=1}^{C} \big(w_{pos} \cdot l_j \log(\hat{l}_j) + (1 - l_j) \log(1 - \hat{l}_j)\big) \tag{13}$$

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

3.3.3. Adversarial loss

Based on the consistent distribution criterion, we introduce the adversarial learning framework (Goodfellow et al., 2014; Ganin et al., 2016) to the CoNet, a classifier D is trained by minimizing the following objective:

$$L_{ad}(b, l_{op}) = -\big(l_{op} \log(\hat{l}_{op}) + (1 - l_{op}) \log(1 - \hat{l}_{op})\big) \tag{14}$$

where l_{op} indicates whether the underlying codes are the original ones (i.e., those encoded by the DHN network, see Fig. 2) or the generated ones (i.e., those from output of the CON sub-network), and $\hat{l}_{op}$ is the predicted value of the classifier D. The classifier D can be thought as a way to construct a learnable loss function for the CON network, such that it assigns a lower penalty to those hashing codes which are generated by the CON network but are not distinguishable from the original codes by the classifier D. In other words, the output of the code operator should have a distribution as close as possible to their original input — in this sense, the adversarial learning method works just like a regularization mechanism to the training objective of the CON network.

In summary, the goal of adversarial learning is to adjust CON's weights so as to weaken the trace of code operators, that is to maximize Eq. (14) with a fixed D. Thus we have two modules playing an adversarial game:

$$\max_{CON} \min_{D} L_{ad}(b, l_{op}) \tag{15}$$

To solve Eq. (15), we reverse the gradient back-propagated from the classifier D. The optimization process of the CON network and the classifier D with gradient descent can be roughly denoted as follows:

$$\begin{aligned} \theta_{\text{CON}}^{t+1} &= \theta_{\text{CON}}^{t} - \lambda_r \cdot \rho \frac{\partial b}{\partial \theta_{\text{CON}}^{t}} \left(-\frac{\partial L_{ad}(b, l_{op})}{\partial b}\right) \\ &= \theta_{\text{CON}}^{t} + \lambda_r \cdot \rho \frac{\partial L_{ad}(b, l_{op})}{\partial \theta_{\text{CON}}^{t}} \\ \theta_{\text{D}}^{t+1} &= \theta_{\text{D}}^{t} - \rho \frac{\partial L_{ad}(b, l_{op})}{\partial \theta_{\text{D}}^{t}} \end{aligned} \tag{16}$$

where θ_{CON}^{t} and θ_{D}^{t} are parameters of CON and D at time t. ρ is learning rate, λ_r controls the step size of CON update.

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

$$\begin{aligned} \mathcal{L} = \sum_{i=1}^{N} \Big(&\sum_{j=1}^{3} L_{cl}(b_{ij}, l_{ij}) + \lambda_1 (L_{cl}(b_{i4}, l_{i4}) + L_{cl}(b_{i5}, l_{i5}) \\ &+ L_{cl}(b_{i6}, l_{i6})) + \lambda_2 L_{tr}(b_{i1}, b_{i2}, b_{i3}) \\ &+ \sum_{j=1}^{6} L_{ad}(b_{ij}, l_{op_{ij}}) \Big) \end{aligned} \tag{17}$$

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

3.4. Optimization

Note that the problem of Eq. (17) is a discrete optimization problem, which is NP-hard to solve directly. To address this issue we use two types of relaxation over the loss function as Wu et al. (2017) and Liu et al. (2016) did. 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. (17) is hence rewritten as follows:

$$\begin{aligned} \mathcal{L} = \sum_{i=1}^{N} \Big\{ &\sum_{j=1}^{3} L_{cl}(h_{ij}, l_{ij}) + \lambda_1 (L_{cl}(h_{i4}, l_{i4}) + L_{cl}(h_{i5}, l_{i5}) \\ &+ L_{cl}(h_{i6}, l_{i6})) + \lambda_2 L_{tr}(h_{i1}, h_{i2}, h_{i3}, l_{i1}, l_{i2}, l_{i3}) \\ &+ \sum_{j=1}^{6} L_{ad}(h_{ij}, l_{op_{ij}}) + \lambda_3 \sum_{j=1}^{3} (\||h_{i,j}| - \mathbf{1}\|_1) \Big\} \end{aligned} \tag{18}$$

where $h_* = h(CNN(I_*))$ for short, $h_{i4} = f_{\oplus}(h_{i1}, h_{i2})$, $h_{i5} = f_{\otimes}(h_{i1}, h_{i2})$, $h_{i6} = f_{\ominus}(h_{i4}, h_{i2})$. $\mathbf{1}$ is a vector with all element one. $\|\cdot\|_1$ denotes L1-norm and λ_3 controls the weight of quantization loss. As this new objective is smooth, the optimization for the whole model is performed using stochastic gradient descent based on Adaptive Moment Estimation (Adam). The optimization of each sub-model can be roughly written as follows:

$$\begin{aligned} \theta_{\text{DHN}}^{t+1} &= \theta_{\text{DHN}}^{t} - \rho \frac{\partial \mathcal{L} \backslash L_{ad}}{\partial \theta_{\text{DHN}}^{t}} \\ \theta_{\text{CON}}^{t+1} &= \theta_{\text{CON}}^{t} + \lambda_r \cdot \rho \frac{\partial L_{ad}}{\partial \theta_{\text{CON}}^{t}} - \rho \frac{\partial \mathcal{L} \backslash L_{ad}}{\partial \theta_{\text{CON}}^{t}} \\ \theta_{\text{D}}^{t+1} &= \theta_{\text{D}}^{t} - \rho \frac{\partial L_{ad}}{\partial \theta_{\text{D}}^{t}} \end{aligned} \tag{19}$$

where $\mathcal{L} \backslash L_{ad}$ is the total loss except L_{ad} term.

4. Experiments

4.1. Experimental settings

4.1.1. Datasets

We validate the proposed CoNet method on four multi-label image datasets, i.e., MIRFLICKR25K (Huiskes and Lew, 2008), VOC2012 (Everingham et al., 2010), NUS-WIDE (Chua et al., 2009) and Microsoft COCO (Lin et al., 2014).

MIRFLICKR25K consists of 25,000 multi-label images. Each image is associated with 24 classes. In our experiments, we discard samples with no tags and remain 20,006.

VOC2012 consists of 22,531 images in 20 classes. Since the ground truth labels of the test images are not available, we only use 11,540 images from its training set. For the above two datasets, we randomly sample 2000 images as the test query set and use the remaining images as the training set.

NUS-WIDE contains 260,648 web images belonging to 81 concepts. We only keep the top 10 most frequent labels and the corresponding 186,577 images. We sample 80,000 images as the training set and 2000 images as the testing set.

Microsoft COCO contains 82,783 training images and 40,504 testing images. Each image associated with 90 categories. After pruning images with no category information, we generate 82,081 training samples and 4956 random test samples.

In the training phase, for the MIRFLICKR25K and VOC2012 datasets, we use their whole training set to train the model, whereas, for the NUS-WIDE and COCO datasets, we separately sample 5000 images from their training sets for training. In the testing phase, we use the training set as the dataset to be retrieved and the testing set as queries.

4.1.2. Evaluation protocol

We perform multi-label retrieval with four kinds of retrieval tasks (one with a single image, three with multiple images), which are described 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 test images with *Union* operation as the query.
- **Task 3.** Retrieve relevant images in the training set using a pair of test images with *Intersect* operation as the query.
- **Task 4.** Retrieve relevant images in the training set using a pair of test images with *Subtract* operation as the query.

We randomly select 500 to 1000 pairs from the test set as the query for tasks 2, 3, 4, and ensure each pair shares at least one label on tasks 3 and 4. We adopt the commonly-used Average Cumulative Gain (**ACG**) (Järvelin and Kekäläinen, 2000), Normalized Discounted Cumulative Gain (**NDCG**), and weighted mean Average Precision (**weighted mAP**) as the performance metric. They are defined as:

$$ACG@p = \frac{1}{p}\sum_{i=1}^{p} r_i \tag{20}$$

$$NDCG@p = \frac{1}{Z}\sum_{i=1}^{p} \frac{2^{r_i}-1}{\log(1+i)} \tag{21}$$

where Z is the ideal $DCG@p$ and calculated from the correct ranking list. $r_i = |l_q \cap l_i|$ denotes the semantic similarity between the ith point and the query. l_q and l_i denote the label set of the query and ith position point.

$$\begin{aligned} mAP_w@p &= \frac{1}{Q}\sum_{q=1}^{Q} AP_w@p(q) \\ AP_w &= \frac{\sum_{p=1}^{M} \Pi(r_p > 0) ACG@p}{M_{r>0}} \end{aligned} \tag{22}$$

where $\Pi() \in \{0,1\}$ is an indicator function and $M_{r>0}$ is the number of relevant data points. p is the length of rank list.

4.2. Implementation details

We implement the proposed CoNet with Tensorflow.[2] Detailed configurations are illustrated in Table 1. During training, the batch size is set to 64, the code length to 32, the learning rate to 0.0001, epochs to 250, the λ_1 to 0.01, the λ_2 to 0.1, the λ_3 to 10^{-5}, the λ_r to 1. The w_{pos} is set to 20 for MIRFLICKR25k, VOC2012, NUS-WIDE, and 30 for MS COCO. The margin parameter m is heuristically set to $2k$ (in case of Euclidean relaxation, since m is set to $\frac{k}{2}$ for Hamming distance) to encourage the codes of dissimilar images to differ in no less than a half of bit length. To make the best use of the limited computational resources, we generate all image triplets online from the shuffled training set after each epoch.

Table 1
The configuration of the deep code operation network.

sub-DHN	
CNN	ResNet-152→2048 / VGG16→4096
Fully connect layer	2048/4096→1024, tanh
Fully connect layer	1024→ hash code length k, tanh
sub-CON (one operation)	
Concatenation	[k, k]
Fully connect layer	$2k \rightarrow$ hash code length k, tanh
Classifier D	
Fully connect layer	$k \rightarrow k$
Fully connect layer	$k \rightarrow 2$, softmax

Because the CNN model in the CoNet method is optionally connected, we firstly investigate the influence of jointly learning of the CNN model. We perform the CNN fine-tuning and the CoNet learning in an end-to-end architecture with SGD optimizer, which is named $\text{CoNet}_{\text{e2e}}$. Then it was compared with $\text{CoNet}_{\text{off}}$, which adopts the off-the-shelf CNN features with Adam optimizer. The CNN models of them are the same as the VGG16 model pre-trained on the ImageNet dataset. And two models are trained with 50 epochs and 0.01 learning rate. The results on retrieval task 1 are shown in Table 2. We see that the $\text{CoNet}_{\text{off}}$ method achieves competitive performance to the $\text{CoNet}_{\text{e2e}}$ method and its computation cost is significantly lower than the $\text{CoNet}_{\text{e2e}}$ method by two orders of magnitude. Because the training examples are limited (5000 samples), the $\text{CoNet}_{\text{e2e}}$ cannot be fully fine-tuned, especially in lower layers, which leads to its inferior performance on several datasets, e.g., VCO2102. Therefore, considering the time cost and good performance, the CoNet method adopts the off-the-shelf CNN features in the following experiments.

Besides, to evaluate the performance between the discrete optimization and the relaxation optimization methods, we optimize $\text{CoNet}_{\text{off}}$ with the discrete optimization method (Li et al., 2017) and name it as $\text{CoNet}_{\text{off}*}$. From Table 2, we observe that the performance of $\text{CoNet}_{\text{off}}$ is comparable with that of $\text{CoNet}_{\text{off}*}$, which demonstrates that the relaxation strategy is valid and will not hurt the quality of the learned codes.

4.3. Parameter sensitivity analysis

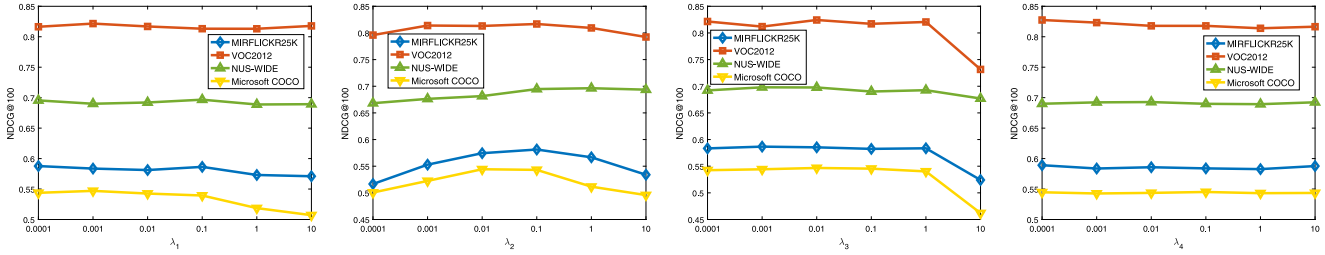
In this section, we analyze the effect of the balance weights λ_1, λ_2, λ_3, λ_r in the proposed method. We set code length to 64 and initially set λ_1, λ_2, λ_3, λ_r to 0.01, 0.1, 0.0001, 1 for all datasets. Then, we separately tune these parameters in the range of {0.0001, 0.001, 0.01, 0.1, 1, 10} with other parameters fixing. The NDCG@100 results of Task 1 on the four datasets are shown in Fig. 3.

We observe that the performance of the CoNet method varies when the parameters are different. Firstly, the proposed method obtains better performance when the λ_1 is smaller than 0.1. Because the λ_1 controls the weight of classification loss of operating codes, a too-large value of it will hurt the discriminative of original codes. Then, we also observe similar results for the parameter λ_3, which is the weighting parameter of quantization loss. A suitable λ_3 is useful to reduce the quantization loss and make the learned feature near to the binary code, while a large λ_3 may lead the optimization process to focus less on preserving similarity.

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

Table 2
Comparison at Top 100 of End-to-End and Off-the-shelf configuration of the CoNet on four datasets.

Method	MIRFLICKR25K			VOC2012			NUS-WIDE			Microsoft COCO			Train time (s)
	NDCG	ACG	mAP_w	NDCG	ACG	mAP_w	NDCG	ACG	mAP_w	NDCG	ACG	mAP_w	
$CoNet_{end2end}$	.4841	2.323	2.362	.7456	1.032	1.052	.6733	1.214	1.222	.4049	1.213	1.324	$\sim 10^4$
$CoNet_{off-the-shelf}$	.4853	2.296	2.338	.7619	1.040	1.063	.6564	1.172	1.187	.4139	1.317	1.346	$\sim 10^2$
$CoNet_{off-the-shelf*}$	.4721	2.251	2.297	.7596	1.035	1.057	.6503	1.164	1.176	.4151	1.315	1.344	$\sim 10^2$



(a) λ_1 (b) λ_2 (c) λ_3 (d) λ_r

Fig. 3. Effect of parameters on the CoNet method. λ_{1-3} balance different loss terms, λ_r controls the weight of CON update in adversarial learning process.

Besides, the proposed method achieves the best performance at a certain value of λ_2. The reason is that a smaller λ_2 may cause the multi-level similarity structure cannot be preserved effectively, while a larger λ_2 may enlarge the noise of similarity and reduce the discriminative of the learned codes. The proposed method is also not much sensitive to the change of the trade-off parameter λ_r, which mainly influences the quality of operating codes (i.e., the performance of tasks 2, 3, and 4).

4.4. Evaluation of different components

To analyze the effectiveness of different components in the proposed CoNet method, we separately remove the L_{ad}, L_{tr} and L_{cl} loss terms with other parameters remained to evaluate their influence on the final performance. These three models are called CoNet-NG, CoNet-NT, and CoNet-NC. Here, NG denotes no adversarial loss L_{ad} term, NC denotes no classification loss L_{cl} term, and NT denotes no multi-level triplet loss L_{tr} term. Fig. 4 shows the NDCG@100 and ACG@100 results of four tasks on all used datasets.

Evaluation of loss terms. From Fig. 4(a), (b), (c) and (d), we see that jointly using the margin-adaptive triplet loss and the classification loss can apparently improve the ranking quality of top-100 relevant items in terms of NDCG and ACG values for the task 1, as it adaptively assigns margins according to the related multilevel similarity and simultaneously encodes rich discriminative information. Also, the CoNet-NT method always performs worse than the CoNet-NC method, and the performance gap is enlarged when the code length increases. This result reveals that the term L_{tr} is more important than L_{cl} term to capture multilevel similarity. However, for task 2, the CoNet-NT method outperforms the CoNet-NC method in most cases. It mainly because the operating codes of the CoNet-NC method contain more noise than the original codes, which makes it hard for the CoNet-NC method to learn correct similarity relation through triplet loss.

Evaluation of adversarial mechanism. From Fig. 4(e), (f), (g), and (h), we observe that the performance of the CoNet is generally similar to that of the CoNet-NG on task 1 and task2, but it is superior on task 3 and task 4. The reason is that the adversarial learning strategy increases the discriminative of the operating codes by weakening the trace of operation and keeps that of the original codes. Notably, in the training phase, we perform twice operations on a pair of codes to accomplish task 4, which means that the operating code of task 4 includes more trace information of operation than other tasks. Thus, the improvement of task 4 with adversarial mechanisms is more apparent in the testing phase. We use t-SNE tools to embed binary-like features of CoNet-NG and CoNet into 2-dimension and visualize their distribution in Fig. 6. We see that the codes from DHN and CON in Fig. 6(b) are more compact than those in Fig. 6(a), which confirms the effectiveness of the adversarial mechanism.

Evaluation of code operation. Besides, to investigate the impact of the code operation network of the CoNet method, we remove the sub-network CON to evaluate its influence on the final performance. The model is named CoNet-NO. Especially, we use logical operations OR, AND, and DIFF for the CoNet-NO method on task 2, task 3, and task 4, respectively, since no code operation network is available. Fig. 5 shows the results. We can see that the CoNet method obtains all-around advantages over the CoNet-NO method on task 1, task 2, and task 4, which demonstrates the effectiveness of the CON sub-network. For task 3, since the 1-bit of code is related to the certain semantic concept, logical AND operation of these codes can treat task 3, the performance of the CoNet-NO method is better than that of the CoNet method, whereas the CoNet-L method (the CoNet method with logical AND operation, see Table 5) can achieve the best performance.

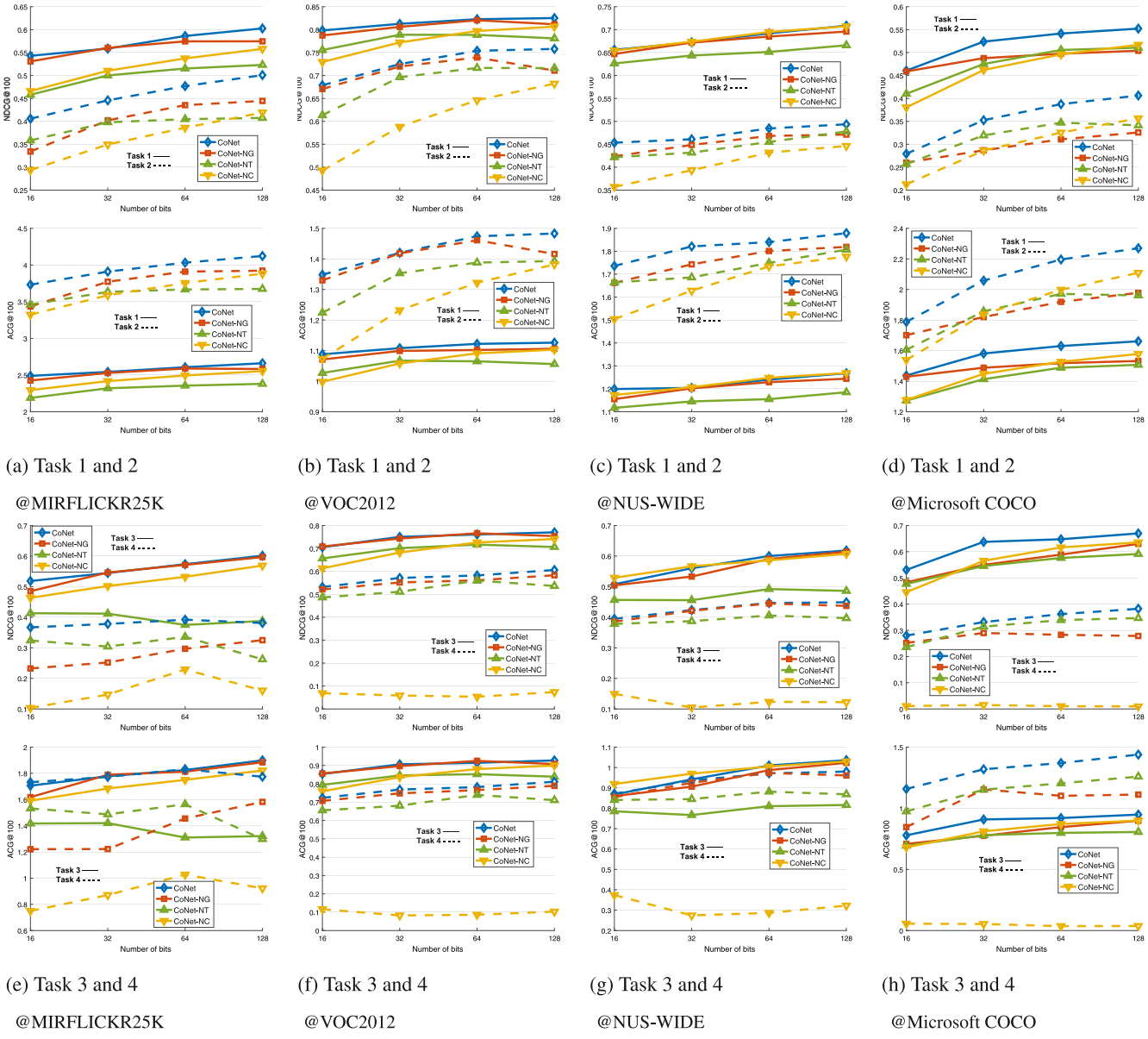
4.5. Experimental result

4.5.1. Comparative methods

We compared the CoNet method with the unsupervised hashing methods **LSH** (Gionis et al., 1999), **SH** (Weiss et al., 2008), **ITQ** (Gong et al., 2013) and the state-of-the-art deep hashing methods **BGAN** (Song et al., 2018b), **MSDH** (Lu et al., 2017), **DRSH** (Zhao et al., 2015), **DMSSPH** (Wu et al., 2017), which are developed for multi-label image retrieval. For a fair comparison, all methods adopt the off-the-shelf 2048-D features from the ResNet-152 (He et al., 2016) which are pre-trained on the ImageNet dataset with Caffe (Jia et al., 2014). The LSH, SH, and ITQ methods were implemented using source codes provided by the authors. To eliminate the effect of the network structures, we implement the MSDH, DRSH, DMSSPH, BGAN and our CoNet methods with the same DHN network structure as Table 1 shows. For unsupervised hashing, BGAN, we construct the similarity matrix with the cosine similarity of semantic labels rather then the original features. All their parameters were set optimally based on the experimental verification. For retrieval tasks 2, 3, and 4, the operation codes of the CoNet-L and other methods are empirically obtained by logical OR, AND, DIFF (DIFF(b_1, b_2)=b_1-AND(b_1, b_2)) operations of two query codes ('−1' is treated as '0'), which is consistent with the fusion manner of semantic labels.

4.5.2. Results on Task 1

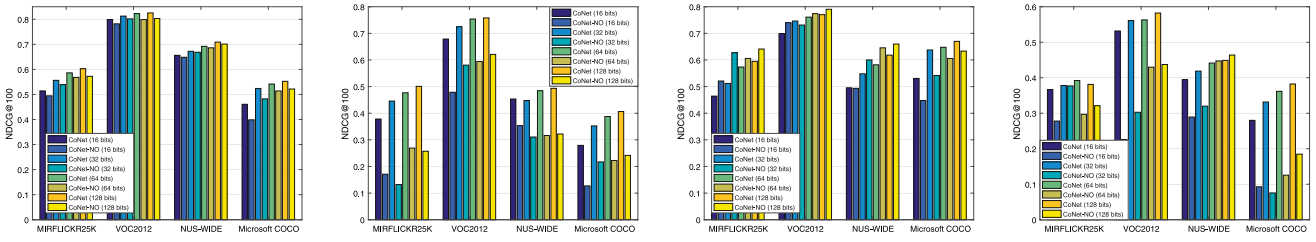
From Table 3, we can observe that the CoNet method substantially outperforms other compared methods with three evaluation criteria on



(a) Task 1 and 2 @MIRFLICKR25K (b) Task 1 and 2 @VOC2012 (c) Task 1 and 2 @NUS-WIDE (d) Task 1 and 2 @Microsoft COCO

(e) Task 3 and 4 @MIRFLICKR25K (f) Task 3 and 4 @VOC2012 (g) Task 3 and 4 @NUS-WIDE (h) Task 3 and 4 @Microsoft COCO

Fig. 4. NDG@100 and ACG@100 of four retrieval tasks with different components and various hashing bits. '-NG' denotes without Classifier D, '-NC' denotes without classification loss L_{cl} term, and '-NT' denotes no multi-level triplet loss L_{tr} term.



(a) Task 1 (b) Task 2 (c) Task 3 (d) Task 4

Fig. 5. NDG@100 on four retrieval tasks of the proposed CoNet and the CoNet-NO (CoNet without codes operation network) with various hashing bits.

all used datasets, e.g., the NDCG@100 results of the CoNet indicate a 2.7% ∼ 4.2% relative increase over the second-best baseline on MS COCO. Compared with the pairwise loss based hashing method DMSSPH, our CoNet method achieves a great improvement, since our CoNet method can exploit richer similarity information in triplets and the discriminative supervised information to capture more accurate multilevel similarity structure.

Besides, it is worth noting that both the CoNet method and the DRSH method learn hashing functions based on the triplet ranking loss, the results in Table 3 shows that the CoNet method gains all-around advantages over the DRSH method. To be more specific, the NDCG@100,

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

Methods	MIRFLICKR25K				VOC2012				NUS-WIDE				COCO			
	16 bit	32 bit	64 bit	128 bit	16 bit	32 bit	64 bit	128 bit	16 bit	32 bit	64 bit	128 bit	16 bit	32 bit	64 bit	128 bit
	NDCG@100															
LSH	.2377	.2635	.3207	.3723	.2586	.3306	.4512	.5450	.3467	.3985	.4830	.5818	.1683	.2370	.3231	.4009
SH	.3238	.3636	.3757	.3849	.5315	.5689	.5662	.5512	.5268	.5591	.5784	.5970	.2809	.3686	.4044	.4244
ITQ	.3577	.4063	.4297	.4482	.5794	.6418	.6694	.6882	.5795	.6028	.6279	.6436	.2761	.3799	.4345	.4674
MSDH	.3950	.4265	.4527	.4602	.6479	.7032	.7081	.6922	.5875	.6128	.6325	.6263	.3338	.4066	.4464	.4721
DRSH	.4396	.4774	.5113	.5301	.6895	.7290	.7572	.7785	.6298	.6639	.6839	.7072	.2994	.3747	.4267	.4553
BGAN	.4452	.4888	5065	.5136	.7370	.7559	.7758	.7796	.6505	.6594	.6654	.6734	.3584	.4645	.4939	.4980
DMSSPH	.4835	.4877	.5004	.5241	.7037	.7136	.7310	.7353	.6120	.6160	.6186	.6157	.4337	.4817	.5053	.5026
CoNet	**.5143**	**.5563**	**.5862**	**.6027**	**.7983**	**.8127**	**.8228**	**.8253**	**.6561**	**.6722**	**.6919**	**.7087**	**.4607**	**.5239**	**.5416**	**.5416**
	ACG@100															
LSH	1.346	1.460	1.681	1.873	.3885	.4763	.6340	.7536	.6670	.7498	.8849	1.043	.6678	.8434	1.074	1.266
SH	1.681	1.805	1.831	1.875	.7303	.7672	.7509	.7239	.9614	1.002	1.035	1.064	.9920	1.164	1.233	1.278
ITQ	1.857	2.016	2.081	2.150	.7959	.8738	.9130	.9362	1.042	1.081	1.122	1.147	.9962	1.222	1.348	1.428
MSDH	1.983	2.086	2.166	2.184	.8733	.9494	.9557	.9368	1.051	1.080	1.123	1.110	1.072	1.258	1.367	1.432
DRSH	2.200	2.316	2.434	2.475	.9448	.9965	1.033	1.064	1.125	1.188	1.224	1.267	1.079	1.248	1.362	1.432
BGAN	2.154	2.315	2.378	2.390	1.001	1.026	1.054	1.060	1.139	1.164	1.178	1.195	1.192	1.434	1.504	1.519
DMSSPH	2.225	2.238	2.292	2.318	.9525	.9688	.9875	.9943	1.077	1.078	1.086	1.082	1.355	1.453	1.509	1.523
CoNet	**2.384**	**2.531**	**2.607**	**2.661**	**1.088**	**1.108**	**1.122**	**1.126**	**1.199**	**1.204**	**1.240**	**1.268**	**1.437**	**1.581**	**1.630**	**1.661**
	mAP$_w$@100															
LSH	1.393	1.531	1.767	1.978	.4442	.5496	.7139	.8357	.6978	.7970	.9341	1.083	.7111	.9015	1.140	1.328
SH	1.767	1.902	1.945	2.001	.7917	.8432	.8484	.8398	.9962	1.042	1.076	1.107	1.035	1.216	1.295	1.347
ITQ	1.886	2.067	2.142	2.213	.8249	.9081	.9527	.9792	1.055	1.100	1.145	1.171	1.024	1.260	1.385	1.470
MSDH	2.018	2.130	2.222	2.256	.8936	.9742	.9862	.9701	1.066	1.091	1.133	1.119	1.091	1.292	1.402	1.472
DRSH	2.216	2.334	2.453	2.504	.9682	1.023	1.055	1.087	1.138	1.197	1.234	1.278	1.085	1.265	1.384	1.462
BGAN	2.178	2.346	2.412	2.428	1.026	1.051	1.079	1.087	1.165	1.174	1.186	1.205	1.211	1.460	1.530	1.547
DMSSPH	2.267	2.294	2.344	2.371	.9729	.9921	1.016	1.023	1.076	1.083	1.091	1.089	1.376	1.476	1.533	1.551
CoNet	**2.425**	**2.569**	**2.647**	**2.699**	**1.108**	**1.125**	**1.140**	**1.142**	**1.180**	**1.214**	**1.247**	**1.279**	**1.464**	**1.599**	**1.657**	**1.671**

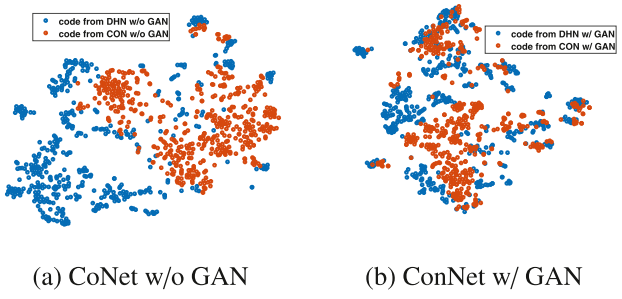


Fig. 6. A visualization of codes. (a) illustrates the distribution of codes from CoNet w/o GAN. (b) illustrates the distribution of codes from CoNet w/ GAN.

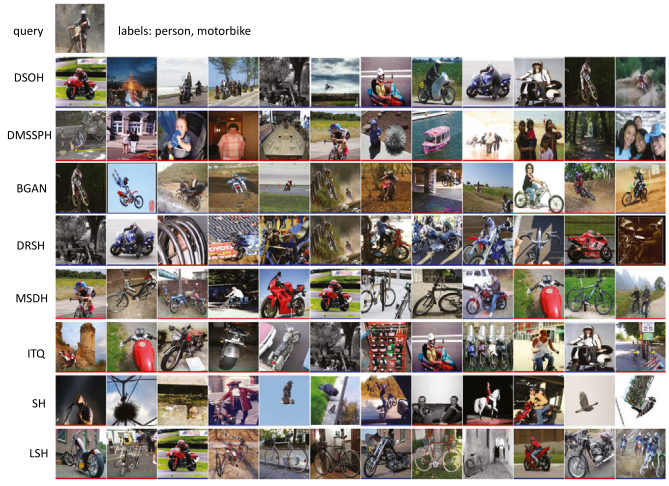


Fig. 7. Retrieval examples of different methods on VOC2012 with 16 bits. Top 12 results for each query are shown. Results with red border denotes $|l_q \cap l_{res}| = 1$, blue denotes $|l_q \cap l_{res}| = 2$ and green denotes $|l_q \cap l_{res}| > 2$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

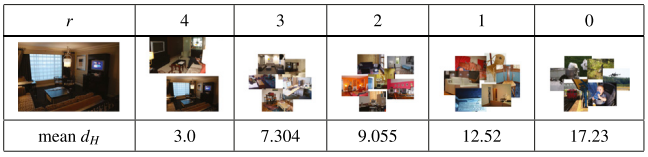


Fig. 8. The consistency of Hamming Distance d_H and Semantic Similarity r on VOC2012 with 32 bits. The second raw shows examples of image that share r semantic labels with the reference example.

ACG@100, and weighted mAP@100 results indicate the relative increase of 8.6%~16.1%, 22.9%~35.8% and 20.9%~37.9% respectively on the MS COCO dataset. This result conclusively demonstrates the benefit of the adaptive margin triplet loss.

Fig. 7 shows the retrieval results of the CoNet method and compared methods. We can see that our CoNet method tends to retrieve more relevant images than other baselines for the query containing certain concepts, i.e., person and motorbike.

To verify the consistency of the hamming distance d_H of learned codes and the multilevel semantic similarity r of labels, we firstly sample a training image as the reference, which holds four tags. We compute the average Hamming distance between this reference and other samples according to their semantic similarity. Fig. 8 shows the results. We can see the d_H and the r exactly satisfy the Eq. (7) requirement.

4.5.3. Results on Task 2

From Table 4, we observe that the CoNet method leads to the superior results with significant improvements (e.g., NDCG: 4.7% ~ 22.6%) over the best-performing compared BGAN method on all datasets. It demonstrates the advantages of the operating codes for complex retrieval task. Because of the non-considering on the consistency between the logical operation of hashing codes and the corresponding fusion manner of the semantic labels, simple OR operation of hashing code of the CoNet-L and other hash methods cannot accomplish the task 2 well.

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

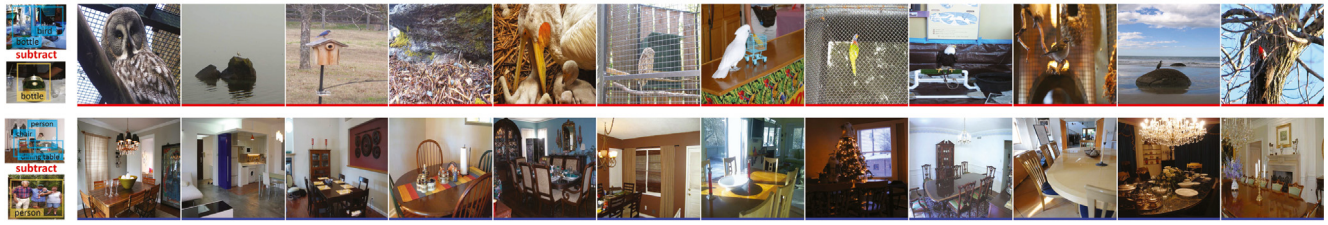
Methods	MIRFLICKR25K				VOC2012				NUS-WIDE				Microsoft COCO			
	16 bit	32 bit	64 bit	128 bit	16 bit	32 bit	64 bit	128 bit	16 bit	32 bit	64 bit	128 bit	16 bit	32 bit	64 bit	128 bit
	NDCG@100															
LSH	.1246	.1135	.1612	.1675	.2022	.2279	.2702	.3349	.1797	.1840	.2184	.2582	.1091	.1279	.1534	.1932
SH	.1104	.1293	.1592	.1500	.2849	.3149	.3201	.3169	.2288	.2467	.2354	.2487	.1250	.1337	.1778	.1802
ITQ	.1697	.1994	.1886	.2147	.3502	.3781	.4227	.4706	.2984	.2430	.2739	.3126	.1243	.1859	.1950	.2227
MSDH	.1461	.1519	.1685	.2094	.3695	.3818	.3654	.4149	.2154	.2722	.2555	.2497	.1509	.1742	.2098	.2057
DRSH	.1729	.1935	.2382	.2965	.3612	.4430	.4844	.5539	.2765	.3125	.3108	.3703	.1274	.1635	.1974	.2306
BGAN	.2021	.2241	.2375	.2328	.4273	.4943	.5502	.5683	.2612	.3142	.3278	.2832	.1934	.2142	.2581	.2622
DMSSPH	.1986	.2347	.2716	.2184	.3994	.4778	.5061	.4682	.2601	.2699	.2801	.2481	.1379	.2332	.2389	.2687
CoNet-L	.1603	.2093	.2578	.2994	.4194	.4518	.6202	.6130	.2291	.2795	.3215	.2897	.2075	.2282	.2743	.2584
CoNet	**.3343**	**.4022**	**.4354**	**.4597**	**.6408**	**.6969**	**.7040**	**.7336**	**.4535**	**.4478**	**.4846**	**.4936**	**.2406**	**.3141**	**.3570**	**.3783**
	ACG@100															
LSH	1.915	1.782	2.228	2.296	.4687	.5331	.6229	.7358	.8266	.9078	1.038	1.134	.8631	.9649	1.076	1.299
SH	1.666	1.889	2.156	2.062	.6456	.6700	.6649	.6553	.9945	1.094	.1049	1.112	.8946	.9345	1.158	1.133
ITQ	2.368	2.584	2.507	2.674	.7979	.8555	.9414	1.005	1.253	1.086	1.189	1.342	.9344	1.323	1.324	1.433
MSDH	2.151	2.137	2.307	2.646	.7916	.8598	.8128	.9240	.9696	1.385	1.095	1.082	1.091	1.194	1.396	1.334
DRSH	2.464	2.659	2.971	3.316	.7803	.9658	1.032	1.158	1.182	1.329	1.322	1.529	1.020	1.246	1.347	1.499
BGAN	2.618	2.721	2.874	2.823	.9678	1.039	1.134	1.167	1.133	1.321	1.354	1.205	1.335	1.402	1.609	1.623
DMSSPH	2.455	2.795	2.976	2.562	.8666	1.002	1.041	.9911	1.088	1.139	1.179	1.051	1.005	1.531	1.478	1.634
CoNet-L	2.288	2.651	3.056	3.268	.8833	.9708	1.264	1.250	1.027	1.198	1.335	1.244	1.452	1.497	1.676	1.655
CoNet	**3.297**	**3.724**	**3.905**	**3.939**	**1.302**	**1.386**	**1.412**	**1.459**	**1.736**	**1.744**	**1.840**	**1.879**	**1.634**	**1.949**	**2.109**	**2.182**
	mAP$_w$@100															
LSH	1.929	1.795	2.283	2.348	.5108	.5762	.6763	.8009	.8450	.9344	1.064	1.164	.8924	.9883	1.116	1.349
SH	1.706	1.943	2.232	2.144	.6917	.7310	.7361	.7459	1.019	1.126	1.082	1.148	.9134	.9696	1.206	1.199
ITQ	2.407	2.628	2.553	2.724	.8398	.8882	.9685	1.041	1.275	1.101	1.204	1.357	.9735	1.133	1.338	1.453
MSDH	2.147	2.155	2.327	2.675	.8091	.8769	.8389	.9389	.9691	1.169	1.097	1.086	1.102	1.201	1.401	1.339
DRSH	2.424	2.640	2.951	3.318	.7946	.9720	1.053	1.185	1.185	1.330	1.320	1.535	1.009	1.217	1.348	1.497
BGAN	2.632	2.713	2.887	2.843	1.001	1.067	1.170	1.207	1.138	1.321	1.352	1.200	1.344	1.423	1.623	1.637
DMSSPH	2.501	2.795	3.072	2.633	.9252	1.071	1.104	1.038	1.123	1.123	1.160	1.048	1.012	1.545	1.478	1.649
CoNet-L	2.322	2.684	3.071	3.311	.9151	.9850	1.301	1.284	1.017	1.201	1.351	1.248	1.446	1.509	1.694	1.670
CoNet	**3.499**	**3.846**	**3.991**	**4.087**	**1.353**	**1.445**	**1.457**	**1.506**	**1.794**	**1.784**	**1.886**	**1.925**	**1.640**	**1.970**	**2.142**	**2.217**

Table 5
NDCG@100, ACG@100 and mAP$_w$@100 of Task 3 on four datasets with different bits length.

Methods	MIRFLICKR25K				VOC2012				NUS-WIDE				Microsoft COCO			
	16 bit	32 bit	64 bit	128 bit	16 bit	32 bit	64 bit	128 bit	16 bit	32 bit	64 bit	128 bit	16 bit	32 bit	64 bit	128 bit
	NDCG@100															
LSH	.2452	.2880	.3351	.3934	.2002	.2518	.4348	.4839	.2962	.2937	.3839	.5128	.1797	.2839	.3017	.4631
SH	.3596	.3653	.3433	.3613	.4523	.4685	.4624	.4313	.4508	.4528	.4449	.4523	.2482	.4194	.4257	.4826
ITQ	.3460	.3865	.4501	.4971	.5009	.5875	.6506	.6605	.4975	.5237	.5764	.5964	.2582	.4452	.5111	.5667
MSDH	.3759	.4486	.5146	.5327	.5510	.6504	.6530	.6340	.5058	.4974	.5428	.5250	.4099	.4998	.5585	.5830
DRSH	.4653	.4988	.5568	.5836	.5665	.6521	.6998	.7310	.5203	.5879	.6691	.6844	.3277	.4035	.4990	.5520
BGAN	.4020	.5164	.5436	.5408	.6696	.6819	.7225	.7391	.5551	.5667	.5971	.5933	.4218	.5703	.5753	.6273
DMSSPH	.4913	.4555	.5003	.5236	.5795	.5972	.6795	.6704	.5238	.5124	.5316	.5131	.5035	.5903	.6273	.6301
CoNet-L	**.5589**	**.5726**	**.6900**	**.6825**	**.7055**	**.7635**	**.7923**	**.8215**	**.6289**	**.5946**	**.6692**	**.6895**	.4904	**.6418**	**.6823**	**.6748**
CoNet	.4643	.5120	.5735	.5951	.6990	.7463	.7610	.7703	.4957	.5483	.5819	.6183	**.5308**	.6376	.6477	.6700
	ACG@100															
LSH	.9321	1.107	1.231	1.368	.2732	.3232	.5362	.5990	.5616	.5339	.6728	.8870	.3107	.4602	.4889	.6833
SH	1.284	1.286	1.184	1.256	.5555	.5654	.5500	.5060	.7860	.7814	.7752	.7759	.4276	.6301	.6032	.6904
ITQ	1.250	1.388	1.528	1.662	.6146	.7134	.7840	.7971	.8818	.9065	.9906	1.009	.4417	.6626	.7572	.8228
MSDH	1.360	1.552	1.722	1.750	.6727	.7829	.7913	.7718	.8882	.8644	.9454	.8984	.6070	.7250	.7981	.8433
DRSH	1.632	1.703	1.870	1.927	.7087	.8012	.8547	.8919	.9238	1.012	1.123	1.147	.5457	.6425	.7577	.8258
BGAN	1.454	1.748	1.813	1.830	.8098	.8366	.8817	.8964	.9627	.9777	1.019	1.019	.6466	.8358	.8414	.9138
DMSSPH	1.675	1.565	1.678	1.743	.7139	.7323	.8273	.8167	.9036	.8861	.9226	.9063	.7614	.8311	.8943	.9000
CoNet-L	**1.832**	**1.845**	**2.141**	**2.135**	**.8645**	**.9282**	**.9601**	**.9909**	**1.073**	**1.013**	**1.132**	**1.169**	.7244	**.9305**	**.9792**	**.9665**
CoNet	1.582	1.678	1.825	1.864	.8474	.8993	.9145	.9276	.8618	.9286	.9756	1.036	**.7788**	.9085	.9201	.9481
	mAP$_w$@100															
LSH	.9778	1.168	1.299	1.465	.3236	.3927	.6105	.6736	.5928	.5795	.7145	.9280	.3501	.5093	.5439	.7356
SH	1.357	1.369	1.276	1.364	.6091	.6424	.6375	.6162	.8285	.8219	.8217	.8303	.4605	.6767	.6491	.7454
ITQ	1.277	1.425	1.584	1.712	.6453	.7473	.8226	.8415	.8961	.9285	1.014	1.030	.4761	.6956	.7903	.8547
MSDH	1.391	1.577	1.765	1.796	.7110	.8142	.8213	.8058	.8967	.8750	.9562	.9109	.6368	.7540	.8275	.8738
DRSH	1.661	1.730	1.892	1.955	.7481	.8374	.8896	.9273	.9385	1.021	1.136	1.158	.5788	.6752	.7832	.8522
BGAN	1.457	1.784	1.839	1.852	.8358	.8721	.9144	.9294	1.006	.9893	1.027	1.026	.6723	.8581	.8690	.9417
DMSSPH	1.644	1.612	1.710	1.781	.7556	.7741	.8686	.8651	.9103	.8937	.9254	.9107	.7809	.8462	.9089	.9223
CoNet-L	**1.872**	**1.877**	**2.184**	**2.172**	**.8883**	**.9582**	**.9929**	**1.021**	**1.085**	**1.022**	**1.138**	**1.182**	.7473	**.9565**	**1.003**	**.9886**
CoNet	1.568	1.692	1.839	1.893	.8622	.9119	.9289	.9415	.8681	.9427	.9842	1.047	**.8039**	.9264	.9396	.9698

(a) Semantic retrieval with hashing codes union operation in the CoNet

(b) Semantic retrieval with hashing codes intersect operation in the CoNet



(c) Semantic retrieval with hashing codes subtract operation in the CoNet

Fig. 9. 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 $|l_q \cap l_{res}| > 2$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Besides, comparing the DRSH and the DMSSPH methods, we find that the DMSSPH method outperforms the DRSH method in most cases (e.g., NDCG values 26,8% v.s. 23.0% on MS COCO with 128 bit), since the DMSSPH method considers preserving multilevel semantic similarity with a certain distance constraint in Hamming space. Although the CoNet method is similar to the DMSSPH method, and both of them use the margin-adaptive strategy in loss function to learn multilevel similarity preserved codes, the proposed CoNet method leads to higher accuracies. The reason can be explained as follows: (1) the exploitation of discriminative supervised information for operation codes in the CoNet method can guide more effective fusion. (2) The comprehensive similarity relationship between the original codes and the fused codes in the triplet is helpful to improve the generalization of the CoNet method.

4.5.4. Results on Task 3

From Table 5, we observe that the CoNet method outperforms other methods by a large margin on the MIRFLICKR, VOC2012, and MS COCO datasets, while the performance of the CoNet method is inferior to the DRSH method on the NUS-WIDE dataset. It mainly because that the tags of images on the NUS-WIDE dataset are more sparse (i.e., most images associate only one tag). It thus is difficult to obtain effective triplets with apparently different multi-level semantic similarity for training. Unlike Task 2, the simple logical AND operation of codes can lead to a superior result in Task 3. For instance, our CoNet-L method achieves the best performance (NDCG, 16 bit: 62.8%, 32 bit: 59.4%, 64 bit: 66.9%, 128 bit: 68.9%) on the NUS-WIDE dataset, while the DRSH method obtains comparable performance (NDCG, 16 bit: 52.0%, 32 bit: 58.8%, 64 bit: 66.9%, 128 bit: 68.4%). This result implies that the 1-bits in learned hash codes of the CoNet and the DRSH methods are actually related to the certain semantic concept, simple logical AND operation can reflect AND fusion of semantics to some degree.

4.5.5. Results on Task 4

Form Table 6, we see similar results to task 2. The CoNet method gains all-around advantages over all the other hashing methods on all used datasets. Specifically, the CoNet method outperforms the best baseline BGAN method by a large margin (8.4% of 16 bits, 5.4% of 32 bits, 5.5% of 64 bits, and 6.9% of 128 bits) in terms of NDCG value on the MS COCO dataset. Because of the non-considering on the consistency between the logical operation of hashing codes and the corresponding fusion of semantic labels, simple DIFF operation of hashing codes for the CoNet-L method and other hash methods cannot accomplish the task 4 well.

In summary, for task 2, task 3 and task 4, the performance of most hash methods are inferior to the CoNet method. This result means that simple logical OR/AND/DIFF operations will hurt the original discriminative of codes and cannot be used for complex retrieval tasks, whereas learning operating codes in the CoNet method is more effective.

We present some retrieval results of task 2, task 3, and task 4 in Fig. 9. From sub-figure (a), we can see the semantic concepts of the query pairs are well fused, and the CoNet method returns meaningful results. For instance, we union images of ‘person’ and ‘cat’ to retrieve images of ’a person holds a cat. ‘It is worth noted that the CoNet method could handle more complex semantic fusion (e.g., uniting images of ’a person are riding a bicycle’ and ‘car’ to retrieve images of ‘a person rides a bicycle in front of a car’). Besides, from sub-figure (b), it can be seen the overlap semantic concepts are used for more precise retrieval. While in sub-figure (c), the semantic concepts of the first query hashing code are partially removed by ‘subtract’ the second query code (e.g., combining images of ‘person, chair and dining table’ and ‘person’ to retrieve images of ‘a dining room with a dining table and several chairs’).

Table 6
NDCG@100, ACG@100 and mAP$_w$@100 of Task 4 on four datasets with different bits length.

Methods	MIRFLICKR25K				VOC2012				NUS-WIDE				Microsoft COCO			
	16 bit	32 bit	64 bit	128 bit	16 bit	32 bit	64 bit	128 bit	16 bit	32 bit	64 bit	128 bit	16 bit	32 bit	64 bit	128 bit
	NDCG@100															
LSH	.1175	.1613	.1767	.1821	.1555	.1241	.1611	.2129	.1833	.1822	.1772	.2809	.0327	.0469	.0635	.1231
SH	.1871	.1970	.1842	.2236	.1816	.2135	.2006	.1873	.2414	.2221	.2461	.2878	.0656	.0977	.1331	.1536
ITQ	.1557	.2121	.2450	.2608	.1026	.1893	.2751	.3568	.2599	.2078	.3100	.3701	.0533	.0760	.1028	.2025
MSDH	.1957	.2371	.2985	.2975	.2068	.2816	.2273	.2285	.2903	.2530	.3176	.2662	.0747	.0911	.1450	.1925
DRSH	.1723	.1889	.2131	.2044	.2142	.2074	.3530	.4200	.2664	.2736	.3878	.4041	.0324	.0610	.0786	.0986
BGAN	.3148	.3540	.3743	.3767	.4523	.4563	.4976	.4945	.3841	.4096	.4398	.4701	.1960	.2769	.3061	.3134
DMSSPH	.2570	.2977	.2896	.3537	.1694	.2485	.3628	.3195	.2465	.2503	.2825	.3222	.1146	.1205	.1987	.2443
CoNet-L	**.4093**	**.4500**	**.4755**	**.4968**	.3244	.4272	.4827	.5246	.3797	.3851	.4145	**.5013**	.1179	.1469	.2186	.2351
CoNet	.3668	.3782	.3920	.3812	**.5318**	**.5610**	**.5631**	**.5823**	**.3947**	**.4187**	**.4414**	.4489	**.2800**	**.3316**	**.3618**	**.3824**
	ACG@100															
LSH	.7610	.9789	1.095	.9996	.2051	.1774	.2408	.2995	.4456	.4451	.3908	.6544	.1650	.3106	.3459	.5243
SH	1.160	1.169	1.063	1.236	.2548	.3103	.2706	.2585	.5846	.5396	.5721	.6483	.3773	.4551	.6161	.6361
ITQ	.9927	1.210	1.326	1.377	.1354	.2314	.3544	.4658	.6022	.5170	.7178	.8207	.2801	.3439	.4348	.8353
MSDH	1.151	1.395	1.620	1.571	.2851	.3964	.3084	.2994	.6635	.5847	.7362	.6195	.4217	.4911	.5820	.8035
DRSH	1.133	1.057	1.265	1.209	.3049	.2665	.4824	.5815	.6417	.6420	.8572	.9043	.1913	.3093	.4177	.4936
BGAN	1.667	1.809	1.889	1.883	.6204	.6163	.6626	.6694	.8688	.9197	.9459	1.026	.9048	1.129	1.212	1.239
DMSSPH	1.450	1.610	1.586	1.845	.2576	.3352	.5387	.4686	.5509	.5814	.6637	.7350	.5736	.5711	.8863	.9512
CoNet-L	**1.965**	**2.102**	**2.178**	**2.239**	.4654	.6004	.6317	.7057	.8590	.8978	.9383	**1.107**	.6414	.7919	1.072	1.024
CoNet	1.732	1.773	1.832	1.774	**.7203**	**.7658**	**.7649**	**.7821**	**.8714**	**.9248**	**.9645**	.9800	**1.158**	**1.319**	**1.370**	**1.439**
	mAP$_w$@100															
LSH	.7837	1.015	1.143	1.061	.2455	.2190	.2865	.3632	.4777	.4766	.4192	.6910	.1945	.3505	.3757	.5824
SH	1.208	1.220	1.123	1.336	.2918	.3708	.3295	.3270	.6205	.5741	.6218	.7047	.4159	.4999	.6778	.7160
ITQ	1.007	1.239	1.375	1.441	.1481	.2675	.3945	.5069	.6129	.5361	.7461	.8488	.3039	.3629	.4588	.8557
MSDH	1.169	1.412	1.658	1.623	.3454	.4249	.3487	.3265	.6803	.6079	.7649	.6421	.4384	.5122	.6042	.8359
DRSH	1.110	1.051	1.252	1.203	.3488	.2923	.5039	.5987	.6585	.6566	.8758	.9151	.2107	.3320	.4387	.5049
BGAN	1.696	1.861	1.938	1.932	.6651	.6694	.7336	.7338	.8923	9286	.9677	1.048	.9429	1.172	1.258	1.287
DMSSPH	1.483	1.657	1.569	1.885	.3209	.3923	.5635	.5189	.5877	.5785	.6547	.7391	.5953	.5881	.9138	.9834
CoNet-L	**2.023**	**2.155**	**2.235**	**2.291**	.5448	.6537	.6952	.7392	**.8973**	.9137	.9460	**1.100**	.6727	.8202	1.109	1.041
CoNet	1.774	1.806	1.859	1.807	**.7499**	**.8030**	**.8002**	**.8209**	.8780	**.9373**	**.9731**	.9857	**1.192**	**1.373**	**1.404**	**1.473**

5. Conclusions

In this paper, we investigate the problem of how to facilitate users' expression of their query intention, in the context of large-scale multi-label image retrieval. For this, we propose a novel deep Code Operation Network (CoNet). The major advantage of the CoNet are two folds: first, it allows users to submit multiple images as query instead of a single one, to express their intent more freely; second, it automatically generates additional queries at the near-semantic level to simulate users' potential query intention, with the help of a set of pre-learned code operations. This essentially provides a natural way for users to express their complex query intention while without the need of stating them explicitly — our system will try to figure it out based on the multiple input images. For this purpose, the CoNet is trained with (1) a newly proposed margin-adaptive triplet loss function, which effectively incorporates the hierarchical similarity structures of the semantic space into the learning of the desired code operations; and (2) an adversary learning-based regularization mechanism, which ensures that the distribution of the generated query codes are not too far from the users' original intention. Extensive experimental results on four popular multi-label datasets show the proposed method can learn useful multilevel semantic similarity-preserving binary codes and achieves state-of-the-art retrieval performance, highlighting the benefit of understanding the query intent of users and reducing the complexity of expressing them. We are currently working on the issue of how to personalize the code operations for different user groups.

Acknowledgments

This work is partially supported by National Science Foundation of China (61976115, 61672280, 61732006), AI+ Project of NUAA, China (56XZA18009), research project no. 6140312020413, Jiangsu Innovation Program for Graduate Education, China (KYCX18_0307), and China Scholarship Council (201906830057).

References

Bai, J., Ni, B., Wang, M., Shen, Y., Lai, H., Zhang, C., Mei, L., Hu, C., Yao, C., 2017. Deep progressive hashing for image retrieval. In: Proc. MM. pp. 208–216.
Çakir, F., He, K., Bargal, S.A., Sclaroff, S., 2017. Mihash: Online hashing with mutual information. In: Proc. ICCV. pp. 437–445.
Cao, Y., Liu, B., Long, M., Wang, J., KLiss, M., 2018. Hashgan: Deep learning to hash with pair conditional wasserstein gan. In: Proc. CVPR. pp. 1287–1296.
Cao, Z., Long, M., Wang, J., Yu, P.S., 2017. Hashnet: Deep learning to hash by continuation. In: Proc. ICCV. pp. 5609–5618.
Chaudhuri, U., Banerjee, B., Bhattacharya, A., 2019. Siamese graph convolutional network for content based remote sensing image retrieval. Comput. Vis. Image Underst. 184, 22–30.
Chen, Z., Xu, Z., Zhang, Y., Gu, X., 2018. Query-free clothing retrieval via implicit relevance feedback. IEEE Trans. Multimedia 20 (8), 2126–2137.
Chua, T.-S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.-T., 2009. Nus-wide: A real-world web image database from national university of Singapore. In: Proc. of ACM Conf. on Image and Video Retrieval (CIVR'09).
Duan, Y., Lu, J., Wang, Z., Feng, J., Zhou, J., 2017. Learning deep binary descriptor with multi-quantization. In: Proc. CVPR. pp. 4857–4866.
Everingham, M., Gool, L., Williams, C.K., Winn, J., Zisserman, A., 2010. The pascal visual object classes (voc) challenge. IJCV 88 (2), 303–338.
Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.S., 2016. Domain-adversarial training of neural networks. J. Mach. Learn. Res. 17, 59:1–59:35.
Gionis, A., Indyk, P., Motwani, R., 1999. Similarity search in high dimensions via hashing. In: International Conference on Very Large Data Bases. pp. 518–529.
Gong, Y., Lazebnik, S., Gordo, A., Perronnin, F., 2013. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. IEEE Trans. Pattern Anal. Mach. Intell. 35 (12), 2916–2929.
Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y., 2014. Generative adversarial networks. CoRR abs/1406.2661.
Gordo, A., Larlus, D., 2017. Beyond instance-level image retrieval: Leveraging captions to learn a global visual representation for semantic retrieval. In: Proc. CVPR. pp. 5272–5281.
He, K., Cakir, F., Bargal, S.A., Sclaroff, S., 2017. Hashing as tie-aware learning to rank. arXiv preprint arXiv:1705.08562.
He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proc. CVPR. pp. 770–778.
Huiskes, M.J., Lew, M.S., 2008. The mir flickr retrieval evaluation. In: ACM International Conference on Multimedia Information Retrieval. pp. 39–43.

Järvelin, K., Kekäläinen, J., 2000. Ir evaluation methods for retrieving highly relevant documents. In: International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 41–48.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R.B., Guadarrama, S., Darrell, T., 2014. Caffe: Convolutional architecture for fast feature embedding. In: Proc. MM. pp. 675–678.

Kim, G., Moon, S., Sigal, L., 2015. Ranking and retrieval of image sequences from multiple paragraph queries. In: Proc. CVPR. pp. 1993–2001.

Lai, H., Yan, P., Shu, X., Wei, Y., Yan, S., 2016. Instance-aware hashing for multi-label image retrieval. IEEE Trans. Image Process. 25 (6), 2469–2479.

Li, C., Deng, C., Li, N., Liu, W., Gao, X., Tao, D., 2018. Self-supervised adversarial hashing networks for cross-modal retrieval. CoRR abs/1804.01223.

Li, Q., Sun, Z., He, R., Tan, T., 2017. Deep supervised discrete hashing. In: Proc. NIPS.

Li, W., Wang, S., Kang, W., 2016. Feature learning based deep supervised hashing with pairwise labels. In: Proc. IJCAI. pp. 1711–1717.

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft coco: Common objects in context. In: Proc. ECCV. pp. 740–755.

Lin, J., Morère, O., Veillard, A., Duan, L., Goh, H., Chandrasekhar, V., 2017. Deephash for image instance retrieval: Getting regularization, depth and fine-tuning right. In: Proc. ICMR. pp. 133–141.

Liong, V.E., Lu, J., Tan, Y., 2018. Cross-modal discrete hashing. Pattern Recognit. 79, 114–129.

Liu, L., Shao, L., Shen, F., Yu, M., 2017a. Discretely coding semantic rank orders for supervised image hashing. In: Proc. CVPR. pp. 5140–5149.

Liu, L., Shen, F., Shen, Y., Liu, X., Shao, L., 2017b. Deep sketch hashing: Fast free-hand sketch-based image retrieval. In: Proc. CVPR.

Liu, H., Wang, R., Shan, S., Chen, X., 2016. Deep supervised hashing for fast image retrieval. In: Proc. CVPR.

Lu, J., Liong, V.E., Zhou, J., 2017. Deep hashing for scalable image search. IEEE Trans. Image Process. 26 (5), 2352–2367.

Luo, Y., Yang, Y., Shen, F., Huang, Z., Zhou, P., Shen, H.T., 2018. Robust discrete code modeling for supervised hashing. Pattern Recognit. 75, 128–135.

Ma, Q., Bai, C., Zhang, J., Liu, Z., Chen, S., 2019. Supervised learning based discrete hashing for image retrieval. Pattern Recognit. 92, 156–164.

Mu, Y., Liu, Z., 2017. Deep hashing: A joint approach for image signature learning. In: Proc. AAAI. pp. 2380–2386.

Qian, X., Tan, X., Zhang, Y., Hong, R., Wang, M., 2016. Enhancing sketch-based image retrieval by re-ranking and relevance feedback. IEEE Trans. Image Process. 25 (1), 195–208.

Salvador, A., Hynes, N., Aytar, Y., Marín, J., Ofli, F., Weber, I., Torralba, A., 2017. Learning cross-modal embeddings for cooking recipes and food images. In: Proc. CVPR. pp. 3068–3076.

Seddati, O., Dupont, S., Mahmoudi, S., 2017. Quadruplet networks for sketch-based image retrieval. In: Proc. ICMR. pp. 184–191.

Song, J., Guo, Y., Gao, L., Li, X., Hanjalic, A., Shen, H.T., 2018a. From deterministic to generative: Multimodal stochastic rnns for video captioning. IEEE Trans. Neural Netw. Learn. Syst. 30 (10), 3047–3058.

Song, J., He, T., Gao, L., Xu, X., Hanjalic, A., Shen, H.T., 2018b. Binary generative adversarial networks for image retrieval. In: Thirty-Second AAAI Conference on Artificial Intelligence.

Song, G., Tan, X., 2017. Hierarchical deep hashing for image retrieval. Front. Comput. Sci. 11 (2), 253–265.

Song, J., Zhang, H., Li, X., Gao, L., Wang, M., Hong, R., 2018c. Self-supervised video hashing with hierarchical binary auto-encoder. IEEE Trans. Image Process. 27 (7), 3210–3221.

Tang, J., Li, Z., Zhu, X., 2018. Supervised deep hashing for scalable face image retrieval. Pattern Recognit. 75, 25–32.

Tang, X., Liu, K., Cui, J., Wen, F., Wang, X., 2012. Intentsearch: Capturing user intention for one-click internet image search. IEEE Trans. Pattern Anal. Mach. Intell. 34 (7), 1342–1353.

Tolias, G., Chum, O., 2017. Asymmetric feature maps with application to sketch based retrieval. In: Proc. CVPR. pp. 6185–6193.

Wang, S., Huang, Z., Xu, X., 2015. A multi-label least-squares hashing for scalable image search. In: Proc. ICDM. pp. 954–962.

Wang, D., Song, G., Tan, X., 2019. Bayesian denoising hashing for robust image retrieval. Pattern Recognit. 86, 134–142.

Weiss, Y., Torralba, A., Fergus, R., 2008. Spectral hashing. In: Proc. NIPS. pp. 1753–1760.

Wu, D., Lin, Z., Li, B., Ye, M., Wang, W., 2017. Deep supervised hashing for multi-label and large-scale image retrieval. In: Proc. ICMR. pp. 150–158.

Xiao, C., Wang, C., Zhang, L., Zhang, L., 2015. Sketch-based image retrieval via shape words. In: Proc. ICMR. pp. 571–574.

Yang, E., Deng, C., Li, C., Liu, W., Li, J., Tao, D., 2018. Shared predictive cross-modal deep quantization. IEEE Trans. Neural Netw. Learn. Syst. 29 (11), 5292–5303.

Yuan, X., Ren, L., Lu, J., Zhou, J., 2018. Relaxation-free deep hashing via policy gradient. In: Proc. ECCV. pp. 141–157.

Zhang, Z., Zou, Q., Wang, Q., Lin, Y., Li, Q., 2018. Instance similarity deep hashing for multi-label image retrieval. CoRR abs/1803.02987.

Zhao, B., Feng, J., Wu, X., Yan, S., 2017. Memory-augmented attribute manipulation networks for interactive fashion search. In: Proc. CVPR. pp. 6156–6164.

Zhao, F., Huang, Y., Wang, L., Tan, T., 2015. Deep semantic ranking based hashing for multi-label image retrieval. In: Proc. CVPR. pp. 1556–1564.

Zhou, W., Li, H., Tian, Q., 2017. Recent advance in content-based image retrieval: A literature survey. arXiv.