Deep Multi-Label Hashing for Large-Scale Visual Search Based on Semantic Graph

Chunlin Zhong[1], Yi Yu[2], Suhua Tang[3], Shin'ichi Satoh[4], and Kai Xing[5]

University of Science and Technology of China[1,5], National Institute of Informatics[2,4], The University of Electro-Communications, Tokyo[3]
chlzhong[1]@mail.ustc.edu.cn, {yiyu[2],satoh[4]}@nii.ac.jp, shtang[3]@uec.ac.jp, kxing[5]@ustc.edu.cn

Abstract. Huge volumes of images are aggregated over time because many people upload their favorite images to various social websites such as Flickr and share them with their friends. Accordingly, visual search from large scale image databases is getting more and more important. Hashing is an efficient technique to large-scale visual content search, and learning-based hashing approaches have achieved great success due to recent advancements of deep learning. However, most existing deep hashing methods focus on single label images, where hash codes cannot well preserve semantic similarity of images. In this paper, we propose a novel framework, deep multi-label hashing (DMLH) based on a semantic graph, which consists of three key components: (i) Image labels, semantically similar in terms of co-occurrence relationship, are classified in such a way that similar labels are in the same cluster. This helps to provide accurate ground truth for hash learning. (ii) A deep model is trained to simultaneously generate hash code and feature vector of images, based on which multi-label image databases are organized by hash tables. This model has excellent capability in improving retrieval speed meanwhile preserving semantic similarity among images. (iii) A combination of hash code based coarse search and feature vector based fine image ranking is used to provide an efficient and accurate retrieval. Extensive experiments over several large image datasets confirm that the proposed DMLH method outperforms state-of-the-art supervised and unsupervised image retrieval approaches, with a gain ranging from 6.25% to 38.9% in terms of mean average precision.

Keywords: Learning based hashing, Deep hashing, Image retrieval, Convolutional neural networks

1 Introduction

With the explosive growth of image contents aggregated in the Internet, it is getting more and more important to search images efficiently over massive databases. Latest research shows that learning based hashing is a promising solution to content-based visual retrieval and search over large-scale databases [1–3] due to their high efficiency in computation and storage. The idea of image hashing is

to map high-dimensional image features to low-dimensional binary hash codes while still preserving semantic similarity between images. In this way, calculating the similarity is easier and quicker based on compact hash codes of images than exhaustive search with high-dimensional features. For example, Hamming distance is usually used to describe image similarity in binary hashing space [4, 5], the smaller the distance is, the more similar two images are.

Existing content-based image hashing methods for visual search mainly can be divided into two categories: data-independent and data-dependent [6, 7]. Data-independent methods generate hash functions randomly without considering any prior information, whose performance is unstable and heavily relies on the qualities of hashing functions. In contrast, data-dependent methods [8] are attracting more attentions due to their better performance, which learn hash functions by exploring the attributes of prior information. Early data-dependent approaches [9, 10] learn hash codes from hand-crafted features of images. Features like GIST and Scale-Invariant Feature Transform (SIFT) are extracted from images and then mapped to compact hash codes. However, the distances in hand-crafted features sometimes cannot well capture the semantic similarity from human perception [11], which limit their performance. Recently, deep learning algorithms have become more and more popular because of their powerful abilities in various multimedia applications such as image retrieval and recognition. Inspired by the advancement of deep learning, researchers have proposed many image hashing algorithms based on convolutional neural networks (CNNs) [12–14]. While the majority of existing deep hashing approaches aim to learn models from single label images, which has limited semantic representation of images.

In this paper, we introduce a novel Deep Multi-Label Hashing (DMLH) framework to realize visual content search on large-scale image databases. The main characteristic of our DMLH is to mine the abundant semantic information hidden in multi-label images and preserve it in hash code and feature vector. This method includes the following three main components: (i) Pre-processing via label clustering. The same concept may be annotated with different labels(such as sea and ocean) by different users. Label clustering is to classify similar labels to the same cluster and provide accurate ground truth for the training step. (ii) Deep learning-based hashing. A CNNs based deep learning framework is constructed to learn feature vectors and compact hash codes simultaneously. (iii) Visual content search based on hashing and refined ranking. In the online stage, the feature vector and hash code of a query image are predicted by the trained model, using which we can retrieve most similar images from large-scale dataset rapidly in two steps.

Our deep hashing architecture for visual content search over large-scale datasets contributes in the following aspects:

1) A semantic graph is proposed to cluster labels to reduce data sparsity meanwhile preserving inner information of images based on the co-occurrence relation among labels. This helps maintain semantic information more sufficiently and generate more accurate ground truth for the training step compared with other descriptors.

2) A novel deep CNNs framework is presented to learn visual contents and compact description of images with multi-label semantic vector as input directly, by fine tuning the pre-trained Caffe ImageNet model, which has powerful ability in learning the semantic similarities between multi-label images.
3) We proposed to rank with high-dimensional feature vector the candidate images found by hash code of the query image, which helps to improve retrieval performance significantly with little computational consumption.

Extensive evaluations on several benchmark datasets confirm that our method achieves high performance in terms of several evaluation metrics compared with other state-of-the-art image retrieval approaches, with a gain ranging from 6.25% to 38.9% in terms of mean average precision.

The rest of this paper is organized as follows: The related work about image retrieval is briefly discussed in section 2. Section 3 introduces our deep hashing-based visual search framework in detail. Experimental setting and results are presented in section 4. Finally, we conclude our paper in section 5.

2 Related Work

In recent years, researchers have proposed many methods for hashing based image retrieval [9, 1, 15, 16]. Here we focus on data-dependent methods. These methods can be divided into three categories: unsupervised hashing, supervised hashing and semi-supervised hashing algorithms.

Unsupervised hashing algorithms, such as K-Means Hashing(KMH) [10], Spectral Hashing(SH) [9] and ITerative Quantization(ITQ) [6], do not consider any label information when learning hash functions. SH is a pioneering and classic image retrieval algorithm, which generates hash codes from query images by non-linear functions and Principal Component Analysis(PCA). KMH performs k-means clustering on data firstly and then uses Hamming distances between cluster indices to indicate the similarity among images.

Considering pre-labeled information when training a model, supervised hashing algorithms [2, 17, 16] can further boost image retrieval performance compared to unsupervised methods. Liu *et al.* [2] proposed a kernel-based supervised hashing model based on limited amounts of information by minimizing the distances of similar image pairs and maximizing the distances of dissimilar pairs. Zhang *et al.* [16] learned image hash functions based on a latent factor model. Lin *et al.* [18] achieved fast supervised hashing and high precision by training boosted decision trees.

Sometimes only little information is available for object retrieval. In this case, semi-supervised hashing approaches have remarkable performance. One of them proposed by Wang *et al.* [4] is designed to learn hash functions through minimizing hash code loss on the labeled data while maximizing variance over the labeled and unlabeled data. Semi-supervised hashing algorithm is useful when labeled data is limited.

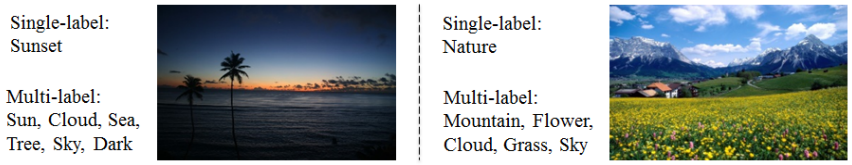

Fig. 1. Example of image label information.

Compared to shallow models, deep learning models have shown significant performance in image classification [19, 20], retrieval [21], and feature extraction [22, 23]. Encouraged by the powerful learning capability of deep learning like CNNs, many frameworks have been proposed by researchers to learn compact hash codes for images to achieve scalable image retrieval [12, 24, 14, 21]. Oquab *et al.* [25] proved that rich image semantic information can be extracted from the mid-levels of CNNs. Erin *et al.* [13] learned a supervised deep hashing(SDH) framework by dividing a training set into positive and negative sample pairs. Kevin *et al.* [26] extracted a hidden layer of CNNs model to generate hash codes for images. This approach is constructed on the successful ImageNet architecture described in [19] and uses single label as supervisal information. However, due to the semantic limitation, single label supervisal information cannot describe the contents of images well and results in semantic information loss, as shown in Fig 1. Lai *et al.* [12] presented an image hashing approach by incorporating a triplet ranking loss function in deep CNNs model to preserve relative similarities between images. Zhao *et al.* [24] constructed a deep learning framework based on semantic ranking with multi-label images. Xia *et al.* [27] embedded the pair-wised image similarity matrix into a deep convolutional model to learn hash functions. These image comparison based hashing algorithms achieve good performance in image retrieval. However, the pre-processing similarity matrixes limit their availabilities in practice. For example, suppose 100k images(a small dataset in large-scale image retrieval) are used in the training phase, the size of pair-wise similarity matrix will be 10 billion, consuming considerable computation and storage resources.

Deep learning algorithms especially CNNs have significant performance in image processing. Inspired by their powerful learning abilities, in this paper, we propose a novel image hashing learning framework DMLH by combining multi-label supervision information and deep CNNs model. Not only semantic information of images is adequately taken into account in our algorithm, but also the practicability and scalability can be ensured in large-scale data environment. The detailed strategy is discussed in the next section.

3 Deep Multi-label Hashing Based Visual Search

As introduced in the previous section, to obtain the discriminating capability of features, the CNNs were used to extract visual features and a hashing layer

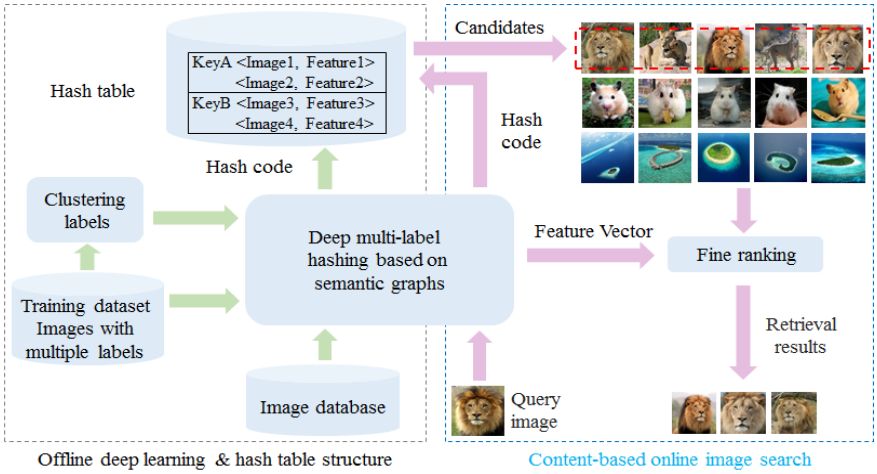

Fig. 2. The framework of DMLH.

was combined together to learn binary codes through single-label supervision. However, an image usually contains a diverse content with different aspects. Obviously, single-label deep hashing limits the semantic descriptions of images. For the purpose of improving semantic representations of images and narrowing semantic gap, the task of multi-label based deep hashing is attracting more attentions. In this work, we mainly focus on developing multi-label deep hashing architecture for efficient and accurate visual search of images on large databases.

Fig 2 shows our framework, which consists of two main stages: offline training and online processing. In the offline stage, a training dataset with multi-label images is used to train a deep multi-label hashing model that predicts hash code and feature vector for an image. To provide accurate ground truth, a pre-processing step, label clustering, is performed. Particularly, the set of diverse image labels is classified to semantic clusters according to label co-occurrences in the annotations of images. Similar labels are classified to the same cluster as far as possible. Then, cluster labels are taken as ground truth to train a multi-label hashing model by aid of CNNs. With the trained model, images in the database are organized in hash tables according to their hash codes. In the online stage, a query image without any label is taken as an input. With the trained deep model, its hash code and feature vector are predicted. The predicted hash code helps to find a small portion of candidates from the database, based on which a ranked list is generated by computing the similarity with feature vector. In the following, we describe the main components of the DMLH framework separately.

3.1 Label Clustering

Multi-label image hashing methods always have superior performance compared with single-label hashing approaches, because multiple labels contain more ac-

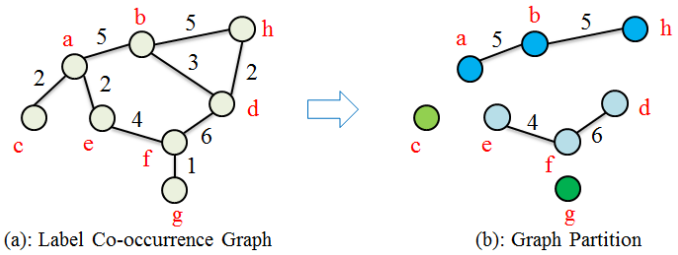

Fig. 3. Label Clustering to preserve semantic similarity.

curate and abundant semantic information compared with single label, such as images shown in Fig 1. However, the same objects often are described by different words due to the diversity of language habits. For example, we can use "people", "human" and "person" to represent "people". What's more, different objects often co-occur in the same image, such as sky and cloud, tree and mountain, which represent another kind of semantic similarity. But each image usually only contains a small number of objects, and is annotated with a small set of labels from a large corpus. Thus, multi-label image hashing approaches are faced with the diversity and sparsity of labels, and computational complexity if a model such as bag-of-words is used to represent the annotation of an image.

For our DMLH framework, in order to reduce computational cost and generate more reliable ground truth for the training step, we propose a novel graph partition algorithm to cluster labels, which contains three parts: *Building Label Co-occurrence Graph*, *Partitioning Semantic Graph*, and *Extracting Image Semantic Vector*.

Building Label Co-occurrence Graph Collaborative Filtering (CF) [28] is one of most efficient algorithms in recommendation systems, and is widely used in industry nowadays. Take item-based CF recommendation as an example. One shopping cart contains a subset of goods, and the similarity between goods is represented by their co-occurrence frequency in the same carts. Two goods are highly similar if they often occur in the same carts. Motivated by the idea of item-based CF, we take labels as goods and images as carts, and use label co-occurrence frequency to represent label semantic similarity. On this basis, we build a label co-occurrence graph (LCG) using labels as nodes and the co-occur frequency of labels as the weight of edges connecting nodes, as shown in Fig 3(a), in which there are 8 labels and 9 similarity values.

Partitioning Semantic Graph According to the idea of item-based CF [28], in LCG, two labels have higher semantic similarity if the edge between them has a larger weight. To extract reliable ground truth and reduce data sparsity, we propose a graph partition algorithm to cluster labels. We hope that labels with semantic similarity can be classified into the same cluster. Our graph partition algorithm is shown in Alg 1. Given a LCG graph (line 2) and K, the number of connected sub-graph (CSG) (line 3), our algorithm removes the edge with the

smallest weight in each iteration until the LCG is divided into K CSGs (line 6-10). Here, a CSG is a sub-graph where from each node there is at least one path to all other nodes. Therefore, a CSG can be built starting from any of its nodes, by iteratively add the neighbors of nodes already found in the CSG. For example, if we set $K = 4$, the LCG in Fig 3(a) will be divided into 4 CSGs after graph partitioning, as shown in Fig 3(b). Removing the edge with the smallest weight, this algorithm keeps the edges with large weights in CSGs, preserving the semantic similarity information as far as possible.

Alg 1: Graph Partition

```
program
1    Output: Sg;    # K clusters of labels
2    Input : LCG;   # Label Co-occurrence Graph
3            K;     # Number of target CSGs
4    var
5        NumOfCSG =  1;
6    While NumOfCSG < K do:
7        edge = FindEdgeWithSmallestWeight(LCG);
8        Remove edge from LCG;
9        NumOfCSG = FindNumberOfConnectedSubGraph(LCG);
10   end while
11   Regard all labels in each CSG as a cluster and add them in Sg;
12   return Sg with K clusters of labels;
end.
```

Extracting Image Semantic Vector The whole label set is clustered into K subsets by graph partitioning Alg 1. In each label subset, there are the highest semantic similarity between each other. Having generating K CSGs, we extract semantic vectors for each multi-label image. each image is represented by a K-dimensional image semantic vector $\boldsymbol{I} \in \{0,1\}^K$, each dimension of which corresponds to one cluster. Given an image with multiple labels, a bit of the vector will be set to 1 if any of its labels appears in the corresponding cluster; Otherwise, this bit is 0. In this way, two images will be represented by similar semantic vectors if there is high similarity between their labels.

The label clustering algorithm not only preserves the semantic similarities between images, but also reduces the sparsity of labels, which help to extract compact and reliable semantic feature vectors from images for the training step and improve the performance of the DMLH model.

3.2 Deep CNNs Based Hashing Learning

Suppose we have an image dataset $\boldsymbol{D} = \{\boldsymbol{x}_i\}_{i=1}^N$ and a class label set $\boldsymbol{L} = \{y_j\}_{j=1}^M$. Each image $\boldsymbol{x}_i \in \boldsymbol{D}$ is associated with $\boldsymbol{S}_i \subseteq \boldsymbol{L}$, a small subset of all labels. $|\boldsymbol{S}_i|$, the size of $\boldsymbol{S}_i$, is 1 in a single-label task and greater than 1 in our multi-label task. The target of image hashing is to learn a mapping as:

$$\boldsymbol{F} : \boldsymbol{D} \rightarrow \{0,1\}^K \tag{1}$$

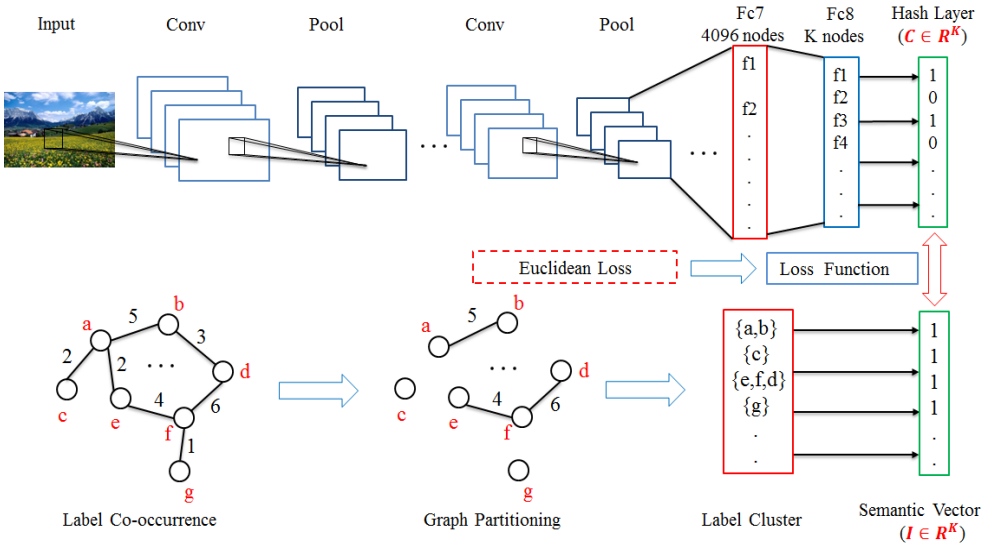

Fig. 4. CNNs hashing structure to learn image hash code and feature vector.

where an input image $\boldsymbol{x}_i$ is encoded into a K-dimensional hash code $\boldsymbol{H}^i = F(\boldsymbol{x}_i)$, and $K \ll M$. The hash codes should preserve the semantic similarities between images after mapping. In specific, if image $\boldsymbol{x}_i$ is similar to $\boldsymbol{x}_j$, the hash codes $\boldsymbol{H}^i$ and $\boldsymbol{H}^j$ should have a small Hamming or Euclidean distance. Otherwise, the distance between $\boldsymbol{H}^i$ and $\boldsymbol{H}^j$ should be large.

Here, we construct a deep learning structure, based on the pre-trained ImageNet model proposed by Krizhevsky *et al.* [19]. The ImageNet model is trained on the 1.2 million images in the benchmark dataset ImageNet, which achieves high accuracy in classifying images into 1,000 object classes. Specifically, the ImageNet model contains five convolutional layers, three pooling layers and two full-connected (Fc) layers. Except for the Fc8 layer, all layers are followed by a ReLU activation layer. The detailed parameter settings can be found in [19].

The structure of our deep multi-label hashing model is shown in Fig 4, which contains two main steps: 1) deep CNNs used to extract semantic representations of images and 2) deep hashing mapping semantic information to compact hash codes. The deep CNNs step contains eight layers, the first seven layers of which have the same structure as ImageNet framework [19]. The number of nodes in the last layer, Fc8, is equal to hash code length K. The deep CNNs step is followed by a nonlinear hashing layer, which is used to convert the output of Fc8 layer to hash codes. Here we select $f(\boldsymbol{x}) = Sigmoid(\boldsymbol{x}) \in \boldsymbol{F}$ as the hash function and all input nodes from Fc8 are mapped to $\{0,1\}$ approximately. $\boldsymbol{H}_i$, the ith bit in the hash code $\boldsymbol{H}$, is generated as follows:

$$\boldsymbol{H}_i = sgn(f(\boldsymbol{x}) - 0.5) = \begin{cases} 1 & f(\boldsymbol{x}) > 0.5, \\ 0 & otherwise. \end{cases} \tag{2}$$

In the training phase, the initial weights of the first seven layers of our model are set the same as in the pre-trained ImageNet model provided by Caffe group, and the connection parameters between Fc7 and Fc8 layers are initialized randomly. Then, we fine-tune the DMLH model on different image datasets to adapt the model to different retrieval requirements. The training goal of our DMLH is to keep the hash codes consistent with the image semantic vectors extracted from images in Section 3.1. The Euclidean distance between hash code $\boldsymbol{H}$ and image semantic vector $\boldsymbol{I}$ is selected as the loss function for back-propagating:

$$Loss = \frac{1}{2K}||\boldsymbol{H} - \boldsymbol{I}||_2^2 = \frac{1}{2K}\sum_{i=1}^{K}(\boldsymbol{H}_i - \boldsymbol{I}_i)_2^2. \tag{3}$$

Where $\boldsymbol{H}, \boldsymbol{I} \in \boldsymbol{R}^K$. Most deep multi-label image hashing methods train models with a pairwise or triple-wise similarity between images, such as [24] and [27]. Except for causing computational explosion, these methods usually need to define complex similarity descriptions between images in the training step. In comparison, our DMLH method uses the image semantic vectors directly, which help our model to extract and preserve visual contents of images sufficiently, achieving high performance and scalability in large-scale visual content search.

3.3 Visual Search by a Combination of Hashing and Fine Ranking

The target of visual content search in this paper is to find top N images most similar to the query image from the database. Our method is performed in two steps for rapid and accurate image retrieval: 1) Deep multi-label hashing is employed for a "coarse" search, narrowing the search region to images having a non-negligible similarity to the query. 2) Ranking of the coarse search is refined by further using feature vector, returning images with a much higher similarity.

Coarse Search Given a query image $\boldsymbol{Q}$, we firstly extract the output of the hash layer from the trained DLMH model and convert it to hash code $\boldsymbol{H}_Q$ with Eq (2). Then, we use $\boldsymbol{H}_Q$ as a key to build a candidate pool $\boldsymbol{P}$ from the hash table of the image database. An image with hash code $\boldsymbol{H}_i$ is added in $\boldsymbol{P}$ if the Hamming distance between $\boldsymbol{H}_Q$ and $\boldsymbol{H}_i$ is less than a pre-defined threshold.

Fine Ranking Studies in [30, 19] show that the Fc6-8 layers of the ImageNet model can preserve sufficient visual information of an input image. The output of Fc7 layer is a 4,096 dimensional feature vector, which preserves more abundant visual information of images compared with hash code. Given one query image $\boldsymbol{Q}$ and each image $\boldsymbol{i}$ its candidate pool $\boldsymbol{P}$, we extract high-dimensional feature vector $\boldsymbol{v}_Q$ and $\boldsymbol{v}_i$ for $\boldsymbol{Q}$ and $\boldsymbol{i}$ from Fc7 layer, respectively. Then, we rank image $\boldsymbol{i} \in \boldsymbol{P}$ based on the Euclidean distance between $\boldsymbol{v}_Q$ and $\boldsymbol{v}_i$ to improve the performance of the proposed DMLH method. The smaller the distance is, the more similar two images are. The top N similar images in the candidate pool $\boldsymbol{P}$ are the final retrieval results for the query image $\boldsymbol{Q}$.

4 Experiment

The performance of DMLH is verified in this section. Our experiments are conducted on a Centos7.2 server, which contains CUDA7.0, Caffe0.14.5 and python2.7. Furthermore, it is configured with E5-1650v4 CPU(3.6GHz), DDR4-2400 Memory(64G) and GeForce GTX TITAN GPU(6144MB).

4.1 Datasets

Two benchmark datasets are chosen to evaluate the performance of our strategy.

NUS-WIDE dataset. This dataset [29] contains 269,648 images collected from the social media sharing website Flickr. Each image is associated with one or several labels in 81 concepts, such as sky, people, and ocean. More specifically, images are marked by a small subset of 5,018 unique semantic tags. For a fair comparison, we follow the works in [12, 27, 7] to use the images associated with the 21 most frequent concepts, where each concept has more than 5,000 images. Our experimental image datasets contain 4,509 tags out of 5,018 unique tags, with approximately all semantic labels. We randomly select 100 images from the top 21 concepts to form query set and 10,500 images are used in the training step, 500 samples for each concept.

MIRFLICKR-25K dataset. This dataset [30] consists of 25,000 images also collected from the Flickr. All images are associated to 24 concepts, such as animals, car and night. 14 stricter concepts are used to label images if a concept is salient in one image. After confirmed, the stricter labels are ignored in our experiments since all images with stricter labels have the concepts before intensified. Further, each image is annotated by a 1,384 dimensional semantic tag vector, in which each tag has appeared more than 20 times in the image dataset. 2,400 images, 100 per concept, are extracted to build the query set and the 9,600 images, 400 images per concept, are used to train the deep CNNs model.

4.2 Evaluation Methods

DMLH algorithm is a two-step visual content search method in which candidate images generated by hash codes are ranked with high-dimensional feature vectors. We evaluate and compare DMLH with several state-of-the-art image hashing methods, including unsupervised methods SH [9], LSH [31], ITQ [6], and supervised methods SDH [32], KSH [2]. For DMLH, we use the raw images as input. For the other baseline methods, one image is represented by a 512-dimensional GIST feature vector. For a fair comparison, we have modified the parameters of baseline methods provided by the original authors to fit with experimental datasets. All approaches use the same ground-truth: a retrieved image is irrelevant if it shares no common concepts with the query image.

4.3 Evaluation Metrics

Three metrics are selected to measure the retrieval performance of different methods: 1) Precision of top N retrieved images, which is calculated as the

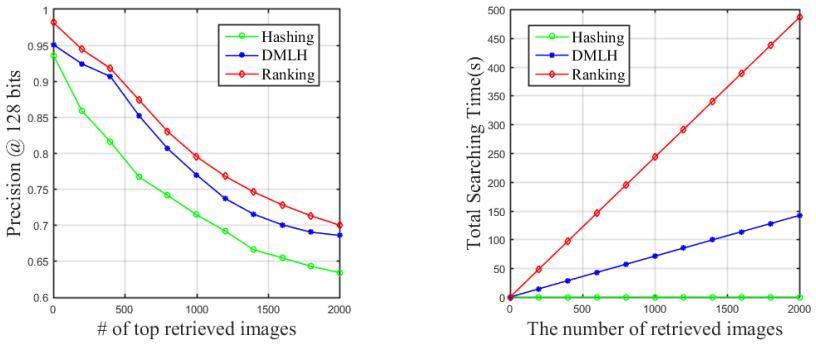

Fig. 5. The equilibrium between image ranking and efficiencies. Hashing means retrieving images only with hash codes. And ranking means that images are retrieved only by high-dimensional feature vectors.

ratio of the number of correctly retrieved images to that of all retrieved images. 2) Recall, which measures the proportion of similar samples that can be retrieved from the image database successfully. 3) Mean Average Precision (MAP), which provides a single figure measure of quality across recall levels.

4.4 Hashing and Ranking

The verification results(Precision and Total Searching Time) of the visual content search method proposed in section 3.3 are shown in Fig 5. Directly ranking images via exhaustive search (Ranking) using high-dimensional feature victors has the best precision because semantic information of images is better preserved. However, it consumes much more time than the other two methods. Hash codes as hashing key (Hashing) help to search images with $O(1)$ time, which achieves fast image search with the time consumption being almost negligible. However, its precision is poor. In our DMLH method, images in the database are assigned to buckets based on the hash codes. In the retrieval stage, candidate images are found by hash codes firstly. Then, high-dimensional feature vectors are used to refine the retrieval results. In this way, its retrieval time is reduced to about 3/10 compared with Ranking while the degradation in precision is suppressed. Therefore, DMLH achieves a better tradeoff than the two extremes (Hashing and Ranking) in terms of retrieval performance (precision) and retrieval time.

4.5 Results on NUS-WIDE

In the offline stage, the semantic graph is constructed on the co-occurrence relations between 4,509 image tags. Then, 128, 96 or 64 subgraphs are generated by $Alg1$. The proposed DMLH is constructed on the opensource Caffe framework. Fig 6 shows the Precision and Recall results of different hashing methods on the NUS-WIDE dataset. As can be seen, compared with unsupervised approaches,

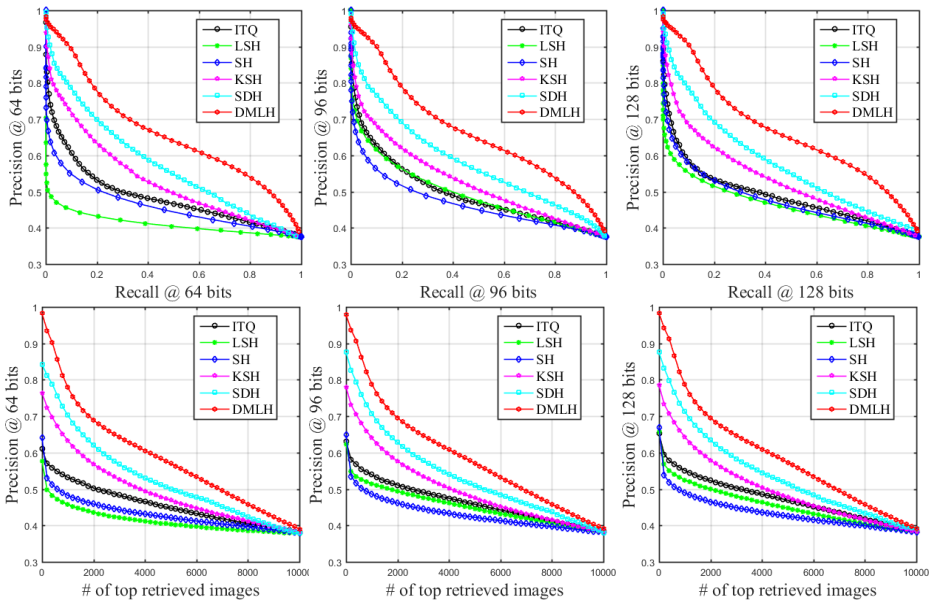

Fig. 6. Comparison of hashing performance of our approach and other hashing methods based on multi-label feature dataset: NUS-WIDE.

Table 1. Comparison of MAP with different methods on NUS-WIDE. We calculate the MAP values within the top 5,000 returned samples.

Method	ITQ [6]	LSH [31]	SH [9]	KSH [2]	SDH [32]	**DMLH**
64 bits	0.452	0.426	0.391	0.493	0.502	**0.543**
96 bits	0.465	0.456	0.414	0.501	0.514	**0.553**
128 bits	0.479	0.466	0.427	0.507	0.528	**0.561**

supervised approaches have better performance by training models under the guidance of a supervised information. And the proposed DMLH significantly outperforms that of other state-of-the-art image retrieval algorithms, regardless of the number of hash bits, 64, 96, 128, used in the hashing. Learning compact image description by a deep CNNs model, our method has better capability to utilize semantic supervised information to capture the visual features of images.

Further, we calculate the MAP values within the top 5,000 retrieved images. Table 1 shows the evaluation results of MAP of different retrieval methods, which indicate that our DMLH has the best ability in retrieving similar images. Its gain over other methods ranges from 6.25% to 38.9% according to Table 1.

4.6 Results on MIRFLICKR-25K

For the MIRFLICKR-25K, we follow the same experimental setup as NUS-WIDE, where the semantic graph is built on the co-occurrence relations among

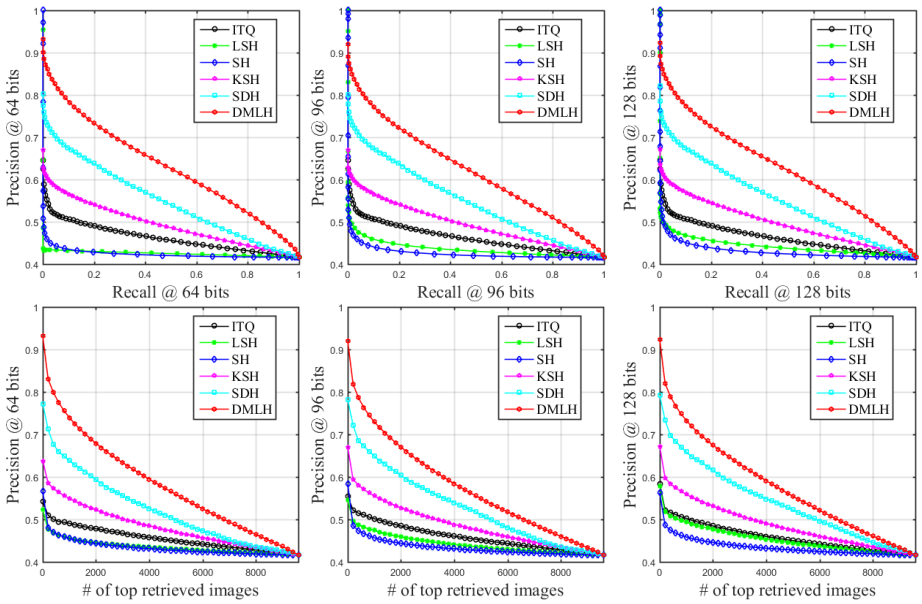

Fig. 7. Comparison of hashing performance of our approach and other hashing methods based on multi-label feature dataset: MIRFLICKR-25K.

Table 2. Comparison of MAP with different methods on MIRFLICKR-25K. We calculate the MAP values within the top 5,000 returned samples.

Method	ITQ [6]	LSH [31]	SH [9]	KSH [2]	SDH [32]	**DMLH**
64 bits	0.426	0.410	0.398	0.439	0.464	**0.501**
96 bits	0.430	0.425	0.410	0.454	0.477	**0.509**
128 bits	0.440	0.431	0.420	0.459	0.489	**0.530**

1,384 semantic tags. Fig 7 and Table 2 show the experimental results of different image retrieval approaches. As can be seen, our DMLH achieves the best performance and is much superior than other hashing methods.

5 Conclusion

In this paper, we proposed a novel visual content search method, DMLH, based on a deep multi-label hashing model. A semantic graph is proposed to preserve the semantic information of images, with which one image is represented by a multi-dimensional semantic vector. Given a raw image without any label, our approach predicts a compact hash code description and high-dimensional feature vector via a carefully designed deep CNNs framework. Compared with other image hashing methods, this method uses multi-label features of images as supervising information, which exploits semantic details of images sufficiently.

Furthermore, DMLH has higher scalability in contrast with other pairwise and triplet-wise deep learning algorithms. Experimental results on two benchmark datasets with multi semantic labels show that our DMLH achieves higher performance than other state-of-the-art image retrieval approaches, whose gain ranges from 6.25% to 38.9% in terms of MAP.

6 Acknowledgments

This research is supported by The National Natural Science Fund under grant 61332004, JSPS KAKENHI grant number 16K16058.

References

1. Norouzi and Blei, "Minimal loss hashing for compact binary codes," in *the 28th international conference on machine learning (ICML-11)*, 2011, pp. 353–360.
2. W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang, "Supervised hashing with kernels," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on.* IEEE, 2012, pp. 2074–2081.
3. Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1891–1898.
4. J. Wang, S. Kumar, and S.-F. Chang, "Semi-supervised hashing for scalable image retrieval," in *CVPR, 2010 IEEE Conference on.* IEEE, 2010, pp. 3424–3431.
5. W. Liu, J. Wang, S. Kumar, and S.-F. Chang, "Hashing with graphs," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011.
6. Y. Gong and S. Lazebnik, "Iterative quantization: A procrustean approach to learning binary codes," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on.* IEEE, 2011, pp. 817–824.
7. W.-J. Li, S. Wang, and W.-C. Kang, "Feature learning based deep supervised hashing with pairwise labels," *arXiv preprint arXiv:1511.03855*, 2015.
8. Y. Zheng, Q. Guo, A. K. Tung, and S. Wu, "Lazylsh: Approximate nearest neighbor search for multiple distance functions with a single index," in *Proceedings of the 2016 International Conference on Management of Data.* ACM, 2016.
9. Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Advances in neural information processing systems*, 2009, pp. 1753–1760.
10. K. He, F. Wen, and J. Sun, "K-means hashing: An affinity-preserving quantization method for learning binary compact codes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2938–2945.
11. J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li, "Deep learning for content-based image retrieval: A comprehensive study," in *Proceedings of the 22nd ACM international conference on Multimedia.* ACM, 2014, pp. 157–166.
12. H. Lai, Y. Pan, Y. Liu, and S. Yan, "Simultaneous feature learning and hash coding with deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3270–3278.
13. V. Erin Liong, J. Lu, G. Wang, P. Moulin, and J. Zhou, "Deep hashing for compact binary codes learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2475–2483.

14. J. Wang, Y. Song, and Leung, "Learning fine-grained image similarity with deep ranking," in *Proceedings of the IEEE Conference on CVPR*, 2014, pp. 1386–1393.
15. M. Norouzi, D. J. Fleet, and R. R. Salakhutdinov, "Hamming distance metric learning," in *Advances in NIPS*, 2012, pp. 1061–1069.
16. P. Zhang, W. Zhang, W.-J. Li, and M. Guo, "Supervised hashing with latent factor models," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM, 2014, pp. 173–182.
17. M. Norouzi, A. Punjani, and D. J. Fleet, "Fast search in hamming space with multi-index hashing," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3108–3115.
18. G. Lin, C. Shen, Q. Shi, A. van den Hengel, and D. Suter, "Fast supervised hashing with decision trees for high-dimensional data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1963–1970.
19. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
20. J. Gao, H. V. Jagadish, W. Lu, and B. C. Ooi, "Dsh: data sensitive hashing for high-dimensional k-nnsearch," in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, 2014, pp. 1127–1138.
21. A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in *European Conference on Computer Vision*, 2014, pp. 584–599.
22. P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.
23. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.
24. F. Zhao, Y. Huang, L. Wang, and T. Tan, "Deep semantic ranking based hashing for multi-label image retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1556–1564.
25. M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1717–1724.
26. K. Lin, H.-F. Yang, J.-H. Hsiao, and C.-S. Chen, "Deep learning of binary hash codes for fast image retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 27–35.
27. R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, "Supervised hashing for image retrieval via image representation learning." in *AAAI*, vol. 1, 2014, p. 2.
28. M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The adaptive web*. Springer, 2007, pp. 325–341.
29. T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Yan-Tao.Zheng, "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)*, July 8-10, 2009.
30. B. T. Mark J. Huiskes and M. S. Lew, "New trends and ideas in visual concept detection: The mir flickr retrieval evaluation initiative," in *MIR '10: Proceedings of the 2010 ACM International Conference on Multimedia Information Retrieval*. New York, NY, USA: ACM, 2010, pp. 527–536.
31. A. Gionis, P. Indyk, R. Motwani *et al.*, "Similarity search in high dimensions via hashing," in *VLDB*, vol. 99, no. 6, 1999, pp. 518–529.
32. F. Shen, C. Shen, W. Liu, and H. Tao Shen, "Supervised discrete hashing," in *Proceedings of the IEEE Conference on CVPR*, 2015, pp. 37–45.