Deep Multiple Instance Hashing for Object-based Image Retrieval

Wanqing Zhao, Ziyu Guan*, Hangzai Luo, Jinye Peng
School of information and technology
Northwestern University, Shaanxi, China
{zhaowanqing,ziyuguan,hzluo,pjy}@nwu.edu.cn

Jianping Fan
Department of Computer Science
UNC-Charlotte, NC28223, USA
jfan@uncc.edu

Abstract

Multi-keyword query is widely supported in text search engines. However, an analogue in image retrieval systems, multi-object query, is rarely studied. Meanwhile, traditional object-based image retrieval methods often involve multiple steps separately. In this work, we propose a weakly-supervised Deep Multiple Instance Hashing (DMIH) framework for object-based image retrieval. DMIH integrates object detection and hashing learning on the basis of a popular CNN model to build the end-to-end relation between a raw image and the binary hash codes of multiple objects in it. Specifically, we cast the object detection of each object class as a binary multiple instance learning problem where instances are object proposals extracted from multi-scale convolutional feature maps. For hashing training, we sample image pairs to learn their semantic relationships in terms of hash codes of the most probable proposals for owned labels as guided by object predictors. The two objectives benefit each other in learning. DMIH outperforms state-of-the-arts on public benchmarks for object-based image retrieval and achieves promising results for multi-object queries.

1 Introduction

Content-based image retrieval (CBIR) has become an active topic in multimedia community since the early 1990s [Smeulders *et al.*, 2000]. Classic CBIR systems take a single query image, and retrieve similar images in a holistic sense from an image repository. However, a user's search interest is usually an object or multiple objects in an image, rather than the entire image. Therefore, object-based (or localized content-based) image retrieval has been defined in [Rahmani *et al.*, 2008], where the user is only interested in a portion of an image. Previous object-based image retrieval methods [Zheng *et al.*, 2006; Rahmani *et al.*, 2008; Li and Liu, 2015] usually involve multiple steps, such as image segmentation, feature extraction and index creation. However, these steps are independent with one another, which would lead to unsatisfactory results. In particular, the errors in the segmentation step will be propagated to the index, and it is difficult to find optimal hand-crafted features for index creation. On the other hand, most existing object-based image retrieval approaches concentrate on querying by a single object. However, users may also want to query about multiple objects, as illustrated in Fig. 1: in the upper example the user is interested in photos containing both human and horses, as in the query; in the lower case the user may want to write a political commentary about Obama and Putin but there is no group photo of them at hand. Due to the lack of ability to detect various objects, previous approaches will incur poor performance when they are directly used to cope with multi-object queries.

Figure 1: Querying about multiple objects.

Recent advances in deep learning have proved that convolutional neural networks (CNNs) trained end-to-end can learn powerful feature representations. In terms of object detection, a number of techniques based on deep CNNs have been proposed [Ren *et al.*, 2015; Liu *et al.*, 2016] and achieved good results on some high-quality public image datasets like ImageNet, PASCAL VOC and MS COCO. However, these approaches necessitate large-scale training data with labels of object locations to learn models to harvest "objectness", whereas the labeling work is very tedious and expensive. It is often the case that we only have image-level object labels without object locations, i.e. weak labels for object detection

*Corresponding author

[Ren *et al.*, 2016]. Multiple Instance Learning (MIL) [Russell *et al.*, 2006] is a particular form of weakly supervised method which can solve the problem above. In MIL, data are labeled in the bag level, where each bag (image in our case) consists of multiple instances (object proposals). MIL assumes that a bag is positive if at least one instance in it is positive, and negative if all the instances in it are negative. The goal of a conventional MIL algorithm is to generate a classifier that will classify unseen bags correctly. Very recently, some CNN-based MIL methods have been proposed to solve the object detection problem by using weak labels [Ren *et al.*, 2016; Kraus *et al.*, 2016]. However, these methods are not suitable to address our problem. First, their objective is to reduce the error rate of object detection, while our task is to construct an object index such that similar objects have similar hash codes. It would result in inefficiency and low accuracy when using the learned CNN features (i.e. outputs of the layer before the prediction layer) from these methods for object retrieval directly, since those features are real vectors and not optimized for similarity search. Second, most of these methods generate large proposals from raw images and feed them into a pretrained CNN to obtain their features. The time cost is unacceptable for image retrieval tasks.

In this paper, a novel weakly-supervised Deep Multiple Instance Hashing (DMIH) framework is proposed for object-based image retrieval. We integrate MIL and hashing learning on the basis of a popular CNN model to build the end-to-end relation between a raw image and the binary hash codes of objects in it. Specifically, we cast the object detection of each object class as a binary MIL problem in which object proposals extracted from trainable multi-scale CNN feature maps (approximately representing sub-areas of different sizes in the input image) are considered as instances and the entire image is treated as a bag. Images containing objects of the class are positive bags, while the remaining ones are negative bags. Together with object predictor learning for each class, a global hash function is also trained to capture the semantic relationships among objects. To this end, we sample image pairs to decrease/increase the embedding distances between same-class/different-class objects in them, where we treat the most probable object proposal assessed by the corresponding object predictor as the object of the class. The whole deep model is optimized via back propagation. In this way, DMIH performs image feature learning, object detection and hash code learning jointly. Object detection helps hashing find correct object proposals and hashing learning provides a regularization for object detection by constraining semantic relationships, thus benefiting each other. After training, we use the hash codes of object proposals judged to be positive by object predictors to represent and index an image.

The main contributions are: (1) we propose a novel learning framework for object-based image retrieval which can well handle multi-object queries; (2) we unify feature learning, hashing learning and MIL based object detection via deep CNNs. The model can effectively generate hash codes of objects in images for retrieval; (3) experiments on three benchmark datasets demonstrate the learned hash codes well preserve the object-level similarity and DMIH outperforms baselines on both single-object and multi-object queries.

2 Related Work

Considering the high dimensionality of images, one critical challenge in CBIR is how to efficiently generate search results. Recently, hashing is recognized as an important technique for fast approximate similarity search. Generally speaking, hashing methods can be categorized into two classes: unsupervised and supervised methods. Unsupervised hashing methods generate compact hash codes by using random projection or training on unlabeled data [Gionis *et al.*, 2000; Weiss *et al.*, 2008; Lee *et al.*, 2010]. The most representative one is Locality-Sensitive Hashing (LSH) [Gionis *et al.*, 2000], which aimed at maximizing the probability that similar data instances are mapped to similar binary codes. In order to search for all groups of partial duplicate images in an image repository, Lee *et al.* [Lee *et al.*, 2010] divided images into multi-scale regions and extracted min-hash [Chum *et al.*, 2008] values for each region independently. Recent studies have shown that using supervised information can boost the performance of binary hash codes. Supervised hashing methods [Kulis and Darrell, 2009; Liu *et al.*, 2012; Zhao *et al.*, 2015] usually incorporate label information into pairwise similarity estimation for training effective hash functions. However, previous hashing methods were focused on mapping whole images or fixed regions of images into the Hamming space. Our work is different from theirs in that our goal is to learn a hash function for objects in images where the objects are detected automatically.

MIL has been leveraged for object-based image retrieval. Rahmani *et al.* [Rahmani *et al.*, 2008] took images as bags and regions in images as instances. They required users to provide a set of query images with positive and negative labels (or via feedback), and then used MIL to train a set of hypotheses online to rank images. Zhang *et al.* [Zhang *et al.*, 2009] integrated active learning into MIL to efficiently obtain bag labels from the query user. Recently, Li and Liu [Li and Liu, 2015] proposed a graph-based MIL framework which constructed two graphs to describe the affinity relationships between images and between regions respectively. The two graphs provided regularization for MIL. Similar to previous work, it also required a set of labeled images from the query user. The major drawback of these methods is that they hinder user experience by asking for labels. Moreover, all the above MIL methods use hand-crafted features and segment raw images to generate instances. These low level features may not well capture the conceptual objects in images.

Recently, researchers explored using MIL and deep learning to address object detection problems. Kraus *et al.* [Kraus *et al.*, 2016] proposed an MIL approach based on CNNs for classifying and segmenting microscopy images. Their work showed that CNNs combined with MIL can be well trained end-to-end using whole microscopy images with image level labels. However, their approach can only deal with fixed-size cells in microscopy images, while objects in natural images can have arbitrary sizes. In comparison, DMIH can capture objects in multi-scales by generating proposals from multi-scale convolutional feature maps. Wu *et al.* [Wu *et al.*, 2015] used BING [Cheng *et al.*, 2014] for generating object proposals from raw images. These proposals were taken as instances

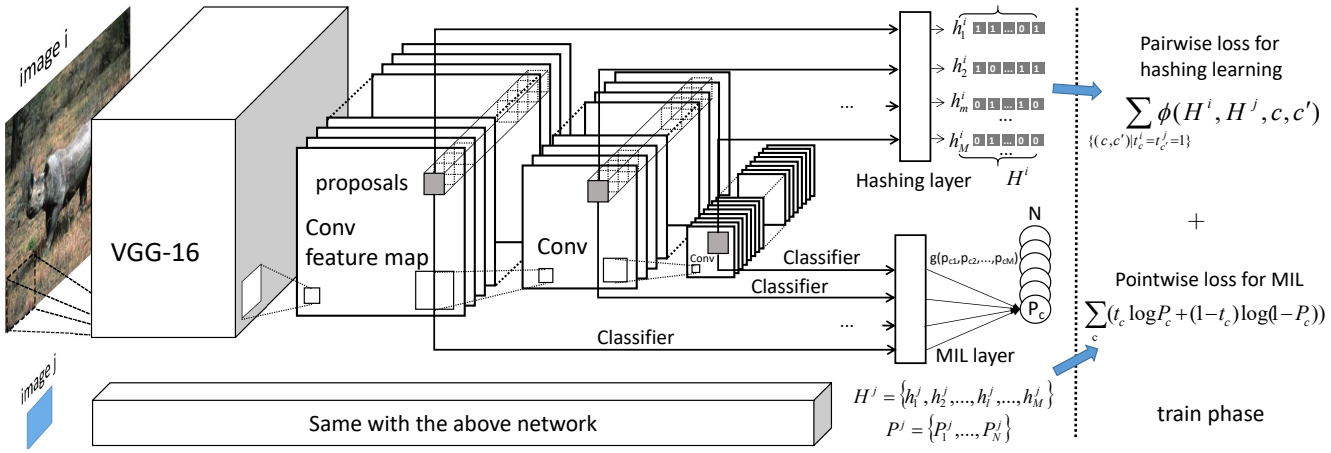

Figure 2: The deep multiple instance hashing learning framework.

and fed into a CNN which was trained in an MIL fashion with image level object labels. Another related work was conducted by Ren *et al.* [Ren *et al.*, 2016], which combined a large pre-trained CNN with MIL for object detection without bounding box annotations. They first produced object proposals on raw images, and each candidate proposal was then encoded by the pre-trained CNN. Finally, the encoded candidate proposals were examined by a SVM classifier which was trained with an MIL objective. However, the above two works aimed to locate and classify objects. The learned high level features were not optimized for retrieval. Second, the methods are inefficient in that each proposal produced from raw images will be encoded by a CNN. The time cost is unacceptable for query processing. In comparison, DMIH generates proposals from high level feature maps, which is more efficient.

3 The Method

3.1 Notations and Architecture Overview

Fig. 2 shows the network architecture. We take VGG-16 [Simonyan and Zisserman, 2015] as our base network to provide high level 2D feature inputs for images. On top of VGG-16 we build multi-scale convolutional layers from which object proposals are extracted via a 3×3 sliding window with stride 1. This scheme for proposal generation has been successfully applied for real-time object detection [Liu *et al.*, 2016]. As in [Liu *et al.*, 2016], the sizes of convolutional filters are set to $3 \times 3 \times p$ where p is the number of feature maps in the previous layer. Finally, object proposals are fed to the MIL/hashing layers for objectness prediction/hash code generation. Note the proposals are generated on feature maps rather than raw images. Their size is much smaller compared to extracting from raw images. This facilitates fast processing of proposals and helps keep the model compact.

Formally, let I_i be the image in the training dataset with index i and $X^i = \{x_1^i, \ldots, x_M^i\}$ be the set of M object proposals extracted for I_i, where $x_m^i \in \mathbb{R}^d$ is the feature vector for proposal m. Let N be the number of object classes. Each image can contain objects from multiple classes. We use $t_c^i \in \{0, 1\}$ to denote whether I_i contains objects from class c. If $t_c^i = 0, \forall c \in \{1, \ldots, N\}$, then I_i has no concerned objects. The top layer of DMIH contains two sub-layers: the hashing layer and the objectness evaluation (MIL) layer. The objectness evaluation layer defines an evaluation function $l_c(\cdot)$ for each class c to judge whether a proposal represents an object of c. The hashing layer maps proposals judged to be objects into the Hamming space through the hash function $f(\cdot)$ for indexing. In the following, we show how to train DMIH.

3.2 Optimization

The parameters of DMIH include $f(\cdot)$, $\{l_c(\cdot)\}_{c=1}^N$ and convolutional filters on top of VGG-16. Firstly, we formulate the objectness evaluation for each class c as an MIL problem where object proposals are treated as instances and images with $t_c^i = 1/t_c^i = 0$ are positive/negative bags, respectively. Given I_i, the evaluation function $l_c(\cdot)$ outputs the probability of a proposal $\mathbf{x}_m^i$ belonging to class c:

$$p_{c,m}^i = l_c(\mathbf{x}_m^i) = \sigma(\mathbf{w}_c^T \mathbf{x}_m^i + b_c) \quad (1)$$

where $\mathbf{w}_c$ and b_c are the parameters of $l_c(\cdot)$ and $\sigma(\cdot)$ is the sigmoid function. After obtaining the probability estimates for all the proposals in X^i, the image level prediction is calculated by applying a global pooling function $g(\cdot)$ over all $p_{c,m}^i$'s as follows

$$P_c^i = g\left(p_{c,1}^i, p_{c,2}^i, \ldots\right) \quad (2)$$

The global pooling function $g(\cdot)$ maps the instance space probabilities to the bag space. Commonly adopted pooling functions include $\max_m(p_{c,m}^i)$, $\text{avg}_m(p_{c,m}^i)$, $\log[1 + \sum_m \exp(p_{c,m}^i)]$, among others. An image would not contain many positive instances. Hence, we adopt $\max(\cdot)$ which means we are concerned with the most probable object proposals. The loss function for the MIL layer is summarized as follows

$$J_{MIL} = -\sum_i \sum_c \left(t_c^i \log P_c^i + (1 - t_c^i) \log(1 - P_c^i)\right) \quad (3)$$

Eq. (3) sums over the loss of each training image. For each image, the loss is a summation of the cross-entropy losses

for all classes. If an image I_i contains no objects (i.e. $t_c^i = 0, \forall c \in \{1, \ldots, N\}$), all the P_c^i's for I_i will be suppressed.

Next, we discuss the training of the hash function $f(\cdot)$. The general idea is that we treat the hashing layer outputs as an embedding Hamming space where we train the semantic relationships between images according to their class labels in terms of the corresponding hash codes. Let $\mathbf{h}_m^i \in \{0,1\}^k$ denote the k bits hash code of object proposal $\mathbf{x}_m^i$ in image I_i, i.e. $\mathbf{h}_m^i = f(\mathbf{x}_m^i)$. We use $H^i = \{\mathbf{h}_1^i, \ldots, \mathbf{h}_M^i\}$ to denote the hash set of the object proposals of image I_i. The intuition is that, if two images I_i and I_j share object labels, we let the corresponding object hash codes be near each other, while we keep objects belonging to different classes away from each other. However, t_c^i's are weak labels at image level. We therefore take the most probable object proposal according to $l_c(\cdot)$ as the object for class c in image I_i with $t_c^i = 1$. For a pair of training images (I_i, I_j), the loss function is defined as:

$$J_{pair-hash}(I_i, I_j) = - \sum_{\{(c,c')|t_c^i = t_{c'}^j = 1\}} \Phi(H^i, H^j, c, c') \tag{4}$$

where the summation is over all possible combinations of class labels contained in I_i and I_j. Motivated by a margin-based loss proposed by [Hadsell *et al.*, 2006], $\Phi(\cdot)$ is defined as:

$$\Phi(H^i, H^j, c, c') = \begin{cases} \mathrm{Dst}(\mathbf{h}^i_{Idx(P_c^i)}, \mathbf{h}^j_{Idx(P_{c'}^j)}), & \text{if } c = c' \\ \max\left(0, \beta - \mathrm{Dst}(\mathbf{h}^i_{Idx(P_c^i)}, \mathbf{h}^j_{Idx(P_{c'}^j)})\right), & \text{otherwise} \end{cases} \tag{5}$$

where $Idx(P_c^i)$ denotes the index of the proposal in I_i with the maximum objectness probability for class c, i.e. $Idx(P_c^i) = \arg\max_m p_{c,m}^i$, and $\mathrm{Dst}(\cdot,\cdot)$ is a distance metric for hash codes. Eq. (5) means that we encourage the most probable same-class objects to be as near as possible, while keeping the most probable different-class objects at least β away. Since the hamming space is discrete and not differentiable, we relax the activation of $f(\cdot)$ to its continuous analogue, the sigmoid function $\sigma(\cdot)$. The approximate hash code of $\mathbf{x}_m^i$ is computed as [Lin *et al.*, 2015]

$$\tilde{\mathbf{h}}_m^i = f(\mathbf{x}_m^i) = \sigma(\mathbf{W}_f^T \mathbf{x}_m^i + \mathbf{b}_f) \tag{6}$$

where $\mathbf{W}_f \in \mathbb{R}^{k \times d}$ and $\mathbf{b}_f \in \mathbb{R}^{k \times 1}$ are the parameters of $f(\cdot)$. The hash codes used for indexing are obtained by binarizing $\tilde{\mathbf{h}}$. $\mathrm{Dst}(\cdot,\cdot)$ is set to be the Euclidean distance accordingly.

Finally, we formulate the joint objective function of DMIH by synthesizing Eqs. (3) and (4):

$$J = \sum_{(i,j)} J_{pair-hash}(I_i, I_j) + \lambda J_{MIL} \tag{7}$$

Such a joint optimization scheme could benefit both objectives: the training of the MIL part can help hashing learning locate the correct proposals; the pair-hashing part provides a regularization for objectness evaluation through underlying learnable convolutional filters.

Training. We employ stochastic gradient descent with image pair sampling to optimize DMIH. The hyper-parameter λ in Eq. (7) controls the balance between the two task losses. β and λ are set to 1.25 and 1 respectively by cross validation. The gradient calculation is straightforward. We omit the details due to space limitation.

3.3 Ranking Criterion

After training, we can then use the model for image indexing and query processing. In particular, hash codes of the object proposals whose objectness probabilities are greater than a certain threshold θ are selected to represent an image. For an image I_i in an image repository, we obtain a number of hash codes according to θ and put them together to form a hash bag $\bar{H}^i$ to index I_i. A user can use one or multiple images as queries. The queries will be fed into DMIH and a set of hash codes will be outputted. Given a set Q of query images, we obtain n hash codes to form the hash bag $\bar{H}^Q = \{\mathbf{h}_1^Q, \mathbf{h}_2^Q, \ldots, \mathbf{h}_n^Q\}$. We design a United Hamming distance $\mathrm{UHammDst}$ between $\bar{H}^Q$ and the hash bag $\bar{H}^i$ of an image I_i in the image repository:

$$\mathrm{UHammDst}(\bar{H}^Q, \bar{H}^i) = \sum_{r=1}^{n} \min_{\mathbf{h}_j^i \in \bar{H}^i} \left\| \mathbf{h}_j^i - \mathbf{h}_r^Q \right\|_1 \tag{8}$$

Images in the repository will be ranked in ascending order by $\mathrm{UHammDst}(\cdot,\cdot)$ and the top ranked images are returned to the user.

4 Experiments

In this section, we evaluate DMIH on both single-object queries (which is the focus of previous work) and multi-object queries, to show its superiority over baseline methods.

4.1 Datasets

SIVAL. It is a benchmark dataset that emphasizes the task of object-based image retrieval. It consists of 25 different categories, and each category includes 60 images. The object can occur anywhere against highly diverse backgrounds in each image.

Pascal VOC 2007 [Everingham *et al.*, 2007]. It contains 9,963 images with 20 different object categories. This dataset is more challenging than SIVAL, because there are more variations in scale, posture and angle. Images in it can contain multiple labels, so we run experiments with multi-object queries on this dataset.

ILSVRC 2013 detection set. This dataset has a similar task and style with PASCAL VOC, but contains more images and categories. It contains nearly 400K images in 200 object categories. We use this dataset to primarily evaluate the runtime efficiency of DMIH and baseline algorithms.

4.2 Settings and Evaluation Measures

We compare DMIH to state-of-the-art hashing methods, including PmH [Lee *et al.*, 2010] and DSRH [Zhao *et al.*, 2015]. PmH generates hash codes for multiple regions for an image. In addition to the hand-crafted features used in [Lee *et al.*, 2010], we also apply PmH on features computed by

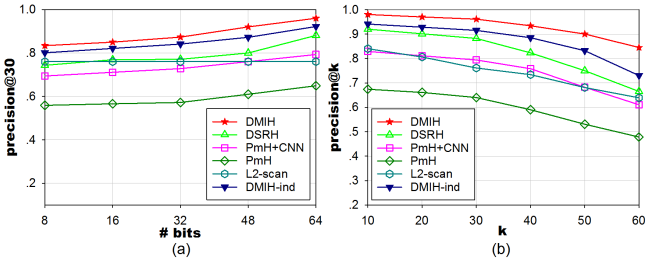

Figure 3: Results on SIVAL; (a) Precision@30 vs. bits; (b) Precision@k vs. k using 64-bit codes.

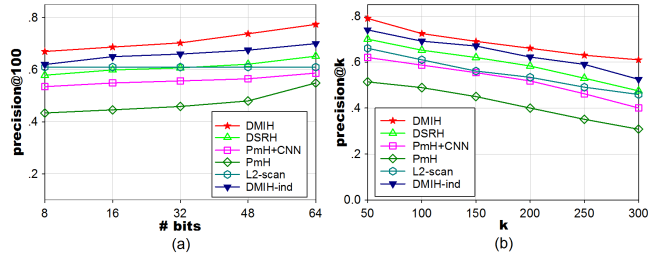

Figure 4: Results on Pascal VOC 2007; (a) Precision@100 vs. bits; (b) Precision@k vs. k using 64-bit codes.

VGG-16 [Simonyan and Zisserman, 2015], denote as PmH-CNN. For DSRH, we first use BING [Cheng *et al.*, 2014] to generate object proposals, and then obtain their hash codes by DSRH [Zhao *et al.*, 2015]. We also compared DMIH to two variants of it. The first one omits the hashing layer and uses Euclidean distance on CNN features ($\mathbf{x}$) for search. It performs deep object detection by MIL in nature and is named ℓ_2-scan. The second one, DMIH-ind, performs MIL and hashing learning separately as two steps. It is used to test whether joint learning leads to better performance. We do not involve traditional object-based retrieval methods since the settings are very different (i.e. they ask for labels from users rather than making use of labeled image datasets). We empirically set the objectness threshold $\theta = 0.7$ for DMIH and its variants. Parameters of PmH and DSRH are set to the best values reported. For fair comparison, we use Eq. (8) as the ranking criterion for all the methods except PmH (it generates a large number of regions for an image). Regarding evaluation metrics, we employ mean average precision (MAP) and precision@k [Lin *et al.*, 2015], where an image is deemed to be relevant to a query if it contains the labels of the query.

Method	TIME(ms)			MAP(%)
	O	H+S	F+O+H+S	64 bits
DMIH	-	-	18.32	74.06
DSRH	6.32	-	27.76	68.12
PmH+CNN	-	18.45	22.97	62.26
PmH	-	19.76	297.43	52.22
ℓ_2-scan	-	-	975.91	66.51

Table 1: Comparison of the average query time and MAP by fixing the code length to 64 bits on ILSVRC 2013.

Multi-object Query	PASCAL VOC 2007 (MAP %)			
	DMIH	DSRH	PmH+ CNN	PmH
bottle + tv	78.6	67.1	61.2	54.3
horse + person	81.2	69.3	64.3	56.6
bus + car	83.1	71.4	66.0	58.4
dog + cat	79.9	64.2	57.8	58.1
bottle + chair + tv	85.7	72.4	69.1	56.1
dog + cat + person	84.3	71.1	67.9	55.2
bus + car + bike	87.1	73.5	70.5	58.4
horse + person + car + dog	91.8	79.2	74.4	61.2
chair + plant + sofa + tv	93.4	80.5	74.9	62.5
Average	85.7	72.6	68.9	57.1

Table 2: Image retrieval results (MAP) for multi-object queries on PASCAL VOC 2007 by fixing the code length to 64 bits.

4.3 Experiments with Single-object Queries

For a single-object query, we assume the whole query image is an object. Each method process query images by taking this prior knowledge into account. We first report results for the SIVAL dataset. We randomly extract 8 images from each object class to form a total of 200 query images. The rest images are used for training and indexing. Fig. 3(a) shows the precision@30 performance of each method when varying hash code length from 8 bits to 64 bits. We find DMIH is consistently better than all the compared methods at each code length. These results illustrate the superior ability of DMIH in extracting and encoding objects from images. In Fig. 3(b), we fix code length to 64 bits and plot precision@k with varying k. Again, DMIH outperforms all the compared methods in all cases. The superiority of DMIH over DMIH-ind demonstrates the usefulness of joint optimization of hashing and MIL objectness prediction. Although the CNN features boost the performance of PmH by an obvious margin, it still performs worse than DMIH. Finally, the mediocre performance of ℓ_2-scan indicates that features trained for object detection are not suitable for retrieval.

We next test on the Pascal VOC 2007 dataset. We randomly sample 50 images from each object category, resulting in a total of 1K query images and 8963 training images. Since each image can contain multiple objects, we only take the object in a query image for the category from which the image was sampled as the query input (the bounding box is given in the dataset). The results for Pascal VOC 2007 are shown in Fig. 4. Since this dataset is much larger than SIVAL with each query having a larger set of relevant images, we report precision@100 and vary k from 50 to 300 in Fig. 4. The observations are similar with those for SIVAL. We also test the performance differences between DMIH and baselines by t-test and find the differences are significant under significance level $\alpha = 0.05$. For the following experiments, we fix hash code length to 64 bits.

For ILSVRC 2013, we randomly extract 100 images from each class as queries. The rest images are used for training and indexing. We use query time to refer to the time

Figure 5: Case studies on PASCAL VOC 2007 for queries with different combinations of object types. Red border denotes false positive.

cost for processing one query, including feature extraction, object/region generation, hash codes calculation and search (abbreviated as "F", "O", "H" and "S" respectively). We report in Table 1 the average query time (and also the retrieval MAP) for each method on ILSVRC 2013. All the methods are run on a PC with NVIDIA GTX 1070 GPU, Inter Core i7-7700 CPU and 16GB memory. Since DMIH is an end-to-end method, we simply report its query time in whole. In DSRH, the time cost of "O" mainly comes from BING (only one proposal generated since the whole image is an object). We use GPUs to only accelerate CNN computation. From Table 1, We can see that DMIH is very efficient compared to baseline methods. This is because DMIH integrates feature extraction, object generation and hash code computation in one deep model. The computation can be well accelerated on GPUs. We omit the results for DMIH-ind since its query time is the same as DMIH. Regarding retrieval performance, DMIH again beats all the other methods.

4.4 Experiments with Multi-object Queries

A user can select one or more images to form a multi-object query. The query images are then processed by each method in the same way as images in the image repository, i.e. generating a bag of hash codes. The returned images should contain different types of objects that appear in the query images. Because most of the images in Pascal VOC 2007 contain multiple objects, it is very suitable for this task. We randomly sample 1K images and randomly combine at most 4 images from them to form queries. An image is judged to be relevant if it contains all types of objects in the query. Table 2 gives the MAP results for different combinations of object types in queries. We can see DMIH outperforms baseline methods by an obvious margin. DMIH also beats DMIH-ind and ℓ_2-scan. We omit them due to space limitation. An interesting phenomenon in Table 2 is that the quality of search results gets improved when using more types of objects as queries. This could be because more types of objects can more clearly express a user's interest, just like multi-keyword queries in text search. Fig. 5 shows top ranked images for queries with different combinations of object types. We can see DMIH generates better results than the baseline methods.

5 Conclusion

In this paper, we propose to construct a deep hashing learning framework for object retrieval in a weakly supervised learning setting. Unlike previous object-based image retrieval methods, our framework builds an end-to-end relation between a raw image and the binary hash codes of objects in it for fast indexing. A joint optimization scheme which integrates feature learning, multiple instance learning and hashing learning is presented for learning the deep model. We demonstrate the superiority of the proposed approach over state-of-the-art methods on both single-object and multi-object retrieval problems on three benchmark datasets. To further speedup retrieval, we will investigate indexing in future work.

Acknowledgments

This research is partly supported by National Science Foundation of China under Grant 61672409,61522206,61373118, National High-Technology Program of China (863 Program, Grant No.2014AA015201), Program for Changjiang Scholars and Innovative Research Team in University (No.IRT13090), and Program of Shaanxi Province Innovative Research Team (No.2014KCT-17).

References

[Cheng *et al.*, 2014] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3286–3293, 2014.

[Chum *et al.*, 2008] Ondrej Chum, James Philbin, Andrew Zisserman, et al. Near duplicate image detection: min-hash and tf-idf weighting. In *British Machine Vision Conference*, 2008.

[Everingham *et al.*, 2007] Mark Everingham, Andrew Zisserman, Christopher KI Williams, Luc Van Gool, Moray Allan, Christopher M Bishop, Olivier Chapelle, Navneet Dalal, Thomas Deselaers, Gyuri Dorkó, et al. The pascal visual object classes challenge 2007 (voc2007) results. 2007.

[Gionis *et al.*, 2000] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. Similarity search in high dimensions via hashing. In *VLDB*, volume 99, pages 518–529, 2000.

[Hadsell *et al.*, 2006] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1735–1742, 2006.

[Kraus *et al.*, 2016] Oren Z Kraus, Jimmy Lei Ba, and Brendan J Frey. Classifying and segmenting microscopy images with deep multiple instance learning. *Bioinformatics*, 32(12):i52–i59, 2016.

[Kulis and Darrell, 2009] Brian Kulis and Trevor Darrell. Learning to hash with binary reconstructive embeddings. In *Advances in Neural Information Processing Systems*, pages 1042–1050, 2009.

[Lee *et al.*, 2010] David C Lee, Qifa Ke, and Michael Isard. Partition min-hash for partial duplicate image discovery. In *European Conference on Computer Vision*, pages 648–662, 2010.

[Li and Liu, 2015] Fei Li and Rujie Liu. Multi-graph multi-instance learning with soft label consistency for object-based image retrieval. In *2015 IEEE International Conference on Multimedia and Expo*, pages 1–6, 2015.

[Lin *et al.*, 2015] Kevin Lin, Huei-Fang Yang, Jen-Hao Hsiao, and Chu-Song Chen. Deep learning of binary hash codes for fast image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 27–35, 2015.

[Liu *et al.*, 2012] Wei Liu, Jun Wang, Rongrong Ji, Yu-Gang Jiang, and Shih-Fu Chang. Supervised hashing with kernels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2074–2081, 2012.

[Liu *et al.*, 2016] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37, 2016.

[Rahmani *et al.*, 2008] Rouhollah Rahmani, Sally A Goldman, Hui Zhang, John Krettek, and Jason E Fritts. Localized content-based image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1902, 2008.

[Ren *et al.*, 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[Ren *et al.*, 2016] Weiqiang Ren, Kaiqi Huang, Dacheng Tao, and Tieniu Tan. Weakly supervised large scale object localization with multiple instance learning and bag splitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):405–416, 2016.

[Russell *et al.*, 2006] Bryan C Russell, William T Freeman, Alexei A Efros, Josef Sivic, and Andrew Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1605–1614, 2006.

[Simonyan and Zisserman, 2015] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Advances in Neural Information Processing Systems*, 2015.

[Smeulders *et al.*, 2000] Arnold WM Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[Weiss *et al.*, 2008] Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. In *Advances in Neural Information Processing Systems*, pages 1753–1760, 2008.

[Wu *et al.*, 2015] Jiajun Wu, Yinan Yu, Chang Huang, and Kai Yu. Deep multiple instance learning for image classification and auto-annotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3460–3469, 2015.

[Zhang *et al.*, 2009] Dan Zhang, Fei Wang, Zhenwei Shi, and Changshui Zhang. Interactive localized content based image retrieval with multiple-instance active learning. *Pattern Recognition*, 43(2):478–484, 2009.

[Zhao *et al.*, 2015] Fang Zhao, Yongzhen Huang, Liang Wang, and Tieniu Tan. Deep semantic ranking based hashing for multi-label image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1556–1564, 2015.

[Zheng *et al.*, 2006] Qing-Fang Zheng, Wei-Qiang Wang, and Wen Gao. Effective and efficient object-based image retrieval using visual phrases. In *Proceedings of the 14th ACM International Conference on Multimedia*, pages 77–80, 2006.