Supervised Hash Coding With Deep Neural Network for Environment Perception of Intelligent Vehicles

Chenggang Yan, Hongtao Xie, Dongbao Yang, Jian Yin, Yongdong Zhang, *Senior Member, IEEE*, and Qionghai Dai, *Senior Member, IEEE*

***Abstract*— Image content analysis is an important surround perception modality of intelligent vehicles. In order to efficiently recognize the on-road environment based on image content analysis from the large-scale scene database, relevant images retrieval becomes one of the fundamental problems. To improve the efficiency of calculating similarities between images, hashing techniques have received increasing attentions. For most existing hash methods, the suboptimal binary codes are generated, as the hand-crafted feature representation is not optimally compatible with the binary codes. In this paper, a one-stage supervised deep hashing framework (SDHP) is proposed to learn high-quality binary codes. A deep convolutional neural network is implemented, and we enforce the learned codes to meet the following criterions: 1) similar images should be encoded into similar binary codes, and vice versa; 2) the quantization loss from Euclidean space to Hamming space should be minimized; and 3) the learned codes should be evenly distributed. The method is further extended into SDHP+ to improve the discriminative power of binary codes. Extensive experimental comparisons with state-of-the-art hashing algorithms are conducted on CIFAR-10 and NUS-WIDE, the MAP of SDHP reaches to 87.67% and 77.48% with 48 b, respectively, and the MAP of SDHP+ reaches to 91.16%, 81.08% with 12 b, 48 b on CIFAR-10 and NUS-WIDE, respectively. It illustrates that the proposed method can obviously improve the search accuracy.**

***Index Terms*— Intelligent vehicles, binary codes, supervised hashing, image retrieval, deep learning.**

Manuscript received February 8, 2017; revised May 22, 2017; accepted June 11, 2017. Date of publication October 4, 2017; date of current version December 26, 2017. This work was supported in part by the National Nature Science Foundation of China under Grant 61771468, Grant 61525206, Grant 61671196, and Grant 61327902, in part by the Zhejiang Province Nature Science Foundation of China under Grant LR17F030006, and in part by the Youth Innovation Promotion Association Chinese Academy of Sciences under Grant 2017209. The Associate Editor for this paper was Q. Wang. *(Corresponding author: Hongtao Xie.)*

C. Yan is with the Institute of Information and Control, Hangzhou Dianzi University, Hangzhou, China (e-mail: cgyan@hdu.edu.cn).

H. Xie and Y. Zhang are with the School of Information Science and Technology, University of Science and Technology of China, Hefei 230026, China (e-mail: xiehongtao@ict.ac.cn; zhyd73@ustc.edu.cn).

D. Yang is with the National Engineering Laboratory for Information Security Technologies, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China, and also with the School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai, China (e-mail: yangdongbao@mail.sdu.edu.cn).

J. Yin is with the School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai, China (e-mail: yinjian@sdu.edu.cn).

Q. Dai is with the Department of Automation, Tsinghua University, Beijing, China (e-mail: qhdai@tsinghua.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2017.2749965

I. Introduction

OVER the past decade, there has been significant research effort dedicated to the development of intelligent driver assistance systems and autonomous vehicles, which is intended to enhance safety by monitoring the on-road environment [1]. Image content analysis as an important environment sensing modality has immensely progressed in recent years.

One of the fundamental problems in image content analysis to efficiently recognize the environment is retrieving relevant contents from a large different scene database [2], which encourages approximate nearest neighbor (ANN) search prosperous [3]. To reduce the computational cost in calculating similarities, hashing techniques have attracted broad attentions in the Big Media research area due to the efficiency of compact binary codes [4], [5]. It aims to construct a series of hash functions to map data points from the original space into compact binary codes and preserve the data structure in the original space. Hashing is a powerful technique for nearest neighbor search with hamming distance computation [6]–[8], because bit-wise XOR operation is performed to calculate the individual similarity, which is advantageous for improving computational efficiency. In addition, the compact binary codes are also beneficial for storage efficiency compared to real-valued representations.

Existing hashing techniques can be classified into two categories: data-independent [9]–[13] and data-dependent [14]–[18]. For the first category, random projections are employed to map data points into a feature space, then binarization are performed. For the second category, various statistical learning techniques are utilized to learn hash functions. In the pipelines of most existing hashing methods, input image is firstly represented by a vector of hand-crafted visual descriptors (e.g., GIST [19], HOG [20]) to capture the image semantics against image noise and redundant information [21]. Secondly, the projection and quantization steps are employed to encode the vector into a binary code. The retrieval performance of conventional hashing methods is limited, mainly resulting from two aspects: on the one hand, the fixed hand-crafted features represent the visual similarities of images rather than the semantic similarities [22]. On the other hand, feature representation and projection are mostly studied as two separated problems, which leads to the suboptimal binary codes generated, as the hand-crafted feature representation is not optimally compatible with the binary codes.

Recent revolution in deep learning [23] shows the impressive feature representation power of Convolutional Neural Network (CNN) [24]–[26], which has been demonstrated by the progress in many visual tasks, such as image classification [24], [27], [28], object detection [29], face recognition [30] and so on [31], [32]. The accomplishments are attributed to the ability of CNN, which can learn the rich mid-level image representation to capture the semantic information [33]. Hashing techniques also benefit from the improvement of CNN to obtain high-quality binary codes with the semantic features of images. Recently, several CNN-based hashing methods have been proposed, such as CNNH [34], DNNH [35], and so on [36]–[40], which have testified the satisfactory performance of binary codes obtained by CNN-based hashing. With the development of CNN, it is necessary to study new algorithms to learn more effective binary codes with less bits and make full use of supervised information to capture more representative features.

In this paper, a novel one-stage supervised deep hashing framework with pairwise labels information (SDHP) is proposed to learn compact binary codes for large-scale image search. Fig. 1 presents the basic idea of the proposed approach. The framework overcomes aforementioned problems of conventional hashing methods and utilizes the feature representation power of CNN to capture the semantic similarities of images. Unlike most existing methods which seek linear projection to map data points into binary codes. In this framework, the design of hash function is based on CNN for learning a nonlinear transformation, and it builds an end-to-end relation between raw image pixels and binary codes for fast retrieval. The optimization of the model is under several constrains at the top layer of the deep network with stochastic gradient descent (SGD) method and back-propagation (BP) algorithm.

The contributions of the paper are summarized as follows:

- A deeper CNN is implemented as the basic network to capture the rich mid-level feature representation, and the pairwise images are organized as the inputs of the network to take advantages of supervised information. In order to preserve the semantic similarities of pairwise images, a pairwise loss function is devised to enforce similar images to map into similar binary codes, and the binary codes of dissimilar images should be as different as possible.
- Due to the fact that the quantization error of the real-valued outputs from Euclidean space to Hamming space is inevitable, and the more evenly distributed binary codes are, the more information can be carried. To obtain high-quality compact binary codes, we define two loss functions: first, the quantization loss function encourages the error from Euclidean space to Hamming space minimized. Second, a even distribution loss function compels the binary codes to be evenly distributed. These two optimization objectives are employed to the hash layer of the deep network.
- In order to further improve the discriminative power of binary codes and make full use of the supervised information, we extend SDHP into SDHP+ by integrating a new layer with the classification information into the deep neural network framework. At the same time, this layer is also enforced to preserve the semantic similarities of pairwise images. The double restrictions make the learned binary codes achieve better search accuracy.

Extensive experimental comparisons are conducted between our method and several state-of-the-art hashing algorithms on two standard image retrieval datasets CIFAR-10 and NUS-WIDE. The MAP of SDHP reaches to 87.67% and 77.48% with 48-bit respectively, and the MAP of SDHP+ can reach to 91.16%, 81.08% with 12 bits, 48 bits on CIFAR-10 and NUS-WIDE respectively. It illustrates that the proposed method can obviously improve the search accuracy, and SDHP+ can achieve better search performance even with less bits. The satisfactory experimental results demonstrate that the proposed method is supposed to be effective for environment perception of intelligent vehicles based on image content analysis.

The rest of this paper is organized as follows. Section II presents some related works on hashing techniques. Section III elaborates the details of the proposed framework and details the optimizing methods of obtaining high-quality binary codes. Section IV shows the extensive experimental results on large-scale real-world image corpus CIFAR-10 and NUS-WIDE. Section V concludes the paper.

II. Related Work

A. Conventional Hashing

Recently, hashing is becoming an important technique for fast approximate nearest neighbor search. Generally speaking, existing hashing methods can be categorized into data-independent and data-dependent methods. Data-independent methods randomly generate hash functions which is independent of any training data. Locality-Sensitive Hashing (LSH) [9] is a typical data-independent method, which uses random linear projections to map data into binary codes. It has been proven that the Hamming distance between two binary codes asymptotically approaches the distance in the original feature space with the code length increasing, which results in the necessary of generating long codes to achieve satisfactory performance.

Data-dependent methods attempt to learn similarity-preserving hash functions from the training data, which can be further divided into unsupervised and supervised methods, depended on whether supervised information (e.g., the class labels of images) is involved. Representative unsupervised exemplars include Spectral Hashing (SH) [14], which obtains balanced binary codes by solving a spectral graph partitioning problem. Wang et al. propose PCA-Hash (PCA-H) [15] which is a data-dependent projection learning method such that each hash function is designed to correct the errors made by the previous one sequentially. Gong and Lazebnik [16] propose an Iterative Quantization (ITQ) method by simultaneously

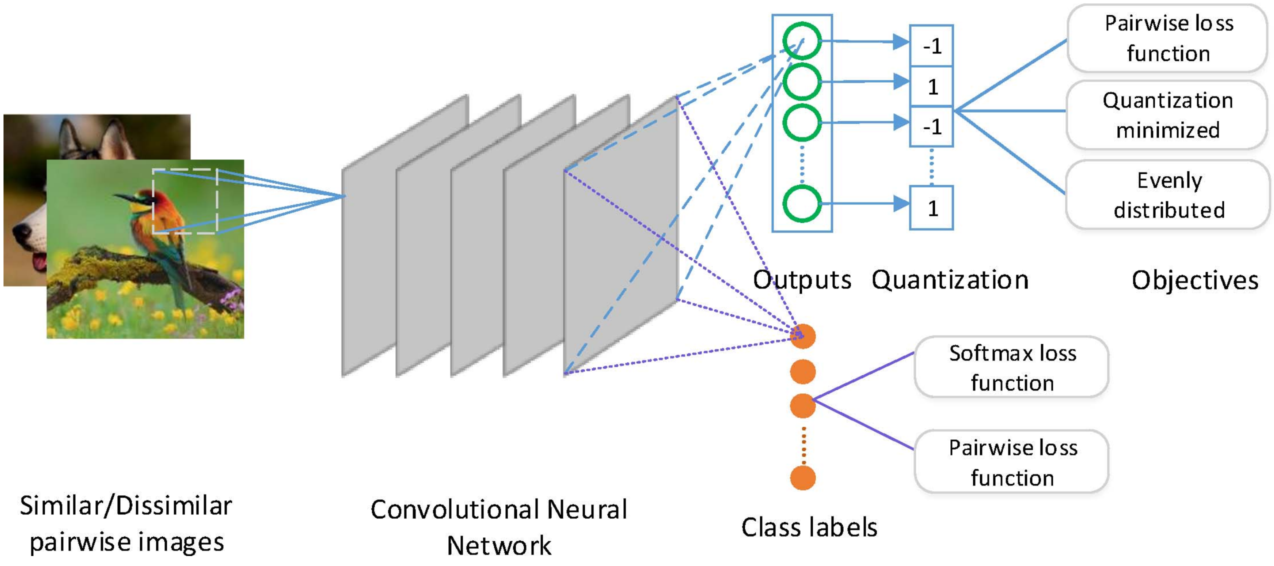


Fig. 1. The framework of the proposed method. The first part is SDHP: the network is trained using image pairs and the label information of images. The learned binary codes should meet the criterions: (a) similar images should be encoded into similar binary codes, and vice versa; (b) the loss of quantization should be minimized; (c) the binary codes should be evenly distributed. The second part is SDHP+: the framework is extended by adding a classification layer to make full use of supervised information.

maximizing the variance of each bit, and minimizing the quantization error of mapping data to the vertices of a binary hypercube.

To obtain semantic similarity for learning hash functions, supervised methods are proposed. In recent years, supervised hashing has attracted more and more attentions because it has better search accuracy than unsupervised methods in many applications. Representative supervised algorithms include Supervised Hashing with Kernels (KSH) [41], Supervised Discrete Hashing (SDH) [17] , Column Sampling based Discrete Supervised Hashing (COSDISH) [18], etc. Liu et al. propose KSH [41] which is a kernel-based method. It learns binary codes by minimizing the Hamming distance between similar pairs and maximizing the distance between dissimilar pairs. SDH [17] leverages label information to obtain binary codes by integrating the generation of hash codes and classifier training, which directly optimize the binary codes to over come the shortcomings of relaxation. COSDISH [18] is a discrete supervised hashing method which can leverage all training data points solving the problem of FastH [42] that cannot utilize all training points due to high time complexity.

B. Deep Hashing

Deep learning learns a hierarchical rich mid-level feature representation that can well capture the semantic information of images. In recent years, CNN-based visual descriptors have been applied on the task of image retrieval. Krizhevsky et al. [24] firstly utilize the feature from the seventh layer of the model for classification, which has achieved impressive performance on ImageNet. Subsequently, more deeper and effective networks are proposed [27], [43]. To our knowledge, semantic hashing [44] is the first using deep learning for hashing. However, the model employs stacked Restricted Boltzmann Machine (RBM) to learn binary codes which is complex and not efficient for practical application. With the boosting studies of Convolutional Neural Network for image classification, CNN-based hashing is researched recently. Xia et al. [34] propose a supervised hashing method CNNH to learn compact binary codes which takes CNN to learn a set of hash functions for the first time, and demonstrate the possibility of CNN applying to hash. However, CNNH is a two-stage method, a matrix-decomposition algorithm applied for learning binary codes in the preprocessing stage. It is unfavorable when the data size is large. Moreover, the learned image feature cannot be used to learn better binary codes due to the separated stages. Subsequently, Lai et al. [35] improve CNNH by proposing a one-stage CNN-based hashing method DNNH for simultaneous feature learning and hashing coding, which enforces the image representation and hash coding to improve each other in a joint learning process. It presents better performance on several benchmarks. However, many hyper-parameters need to be adjusted in this model for better performance.

III. Approach

In this section, the proposed framework is detailed as illustrated in Fig. 1. In order to incorporate feature representation learning and hash coding into an end-to-end framework, we bring up a one-stage supervised hashing method based on deep learning. The learned binary codes are enforced to meet the following criterions: (a) to preserve the semantic relationship of pairwise images, similar images should be encoded into similar binary codes and vice versa; (b) the quantization error from Euclidean space to Hamming space should be minimized; (c) the learned codes should be evenly distributed to carry more information. The pipeline of our method includes three steps: firstly, the training images are organized to pairs, and we train the network with pairwise images and labels information. Secondly, the parameters of the network are optimized with aforementioned criterions. Thirdly, quantization is employed to generate the binary codes from the real-valued outputs of the network. The method for learning compact binary codes is described in detail as follows.

A. Deep Architecture

The training process of the deep architecture includes two main components: the first is network initialization, and the second is optimization. Considering that many famous models have been proposed for classification, and it is demonstrated the effectiveness of these CNN models. The networks have been transferred to many visual tasks which have achieved great success. In this paper, for the first initialization component, GoogLeNet [27] is applied to hashing as the basic framework for the first time, which is pre-trained on the large-scale ImageNet dataset [24]. The dataset contains more than 1.2 million images categorized into 1,000 object classes. Therefore, we mainly focus on the design of hashing layer to adapt to hashing task. The classification layer is replaced with a new fully connected layer with q units, and each unit is associated with one bit, which is enforced to learn approximate binary codes. For the second component, the network for hashing is fine-tuned on the specific image retrieval datasets with stochastic gradient descent (SGD) method and back-propagation (BP) algorithm. The details of optimizing the network will be described in the next part.

B. Formulation

To train the network for hashing, several optimization objectives are devised to obtain high-quality binary codes. The proposed method organizes the training images into pairwise samples. Assuming Ω to be the image space, for a pair of images I_1 and I_2, the goal is to map the images from original space into Hamming space: $\Omega \rightarrow \{+1, -1\}^q$. Each image is represented by a q-bit binary code, hence the binary codes of I_1, I_2 are defined as b_1, b_2.

We enforce every image and its latter image in a batch to be a pair, therefore suppose the number of the images in a batch is n, the number of pairs is $C_n^2 = \frac{n!}{(n-2)!2!}$. The inputs of the network are organized into pairwise images. Meanwhile, a pairwise loss function is devised to enforce similar images mapping into similar binary codes, and the Hamming distance between the binary codes of dissimilar images should be as large as possible. The loss function can preserve the semantic relationship of pairwise images, which can be written as:

$$W_1(b_1, b_2) = \begin{cases} \frac{1}{2}H(b_1, b_2) & S = 1 \\ \frac{1}{2}max(t - H(b_1, b_2), 0) & S = 0 \end{cases}$$
$$s.t.\ b_i \in \{-1, +1\}^q,\ i \in \{1, 2\},$$
$$S = \begin{cases} 1 & I_1\ and\ I_2\ are\ semantically\ similar \\ 0 & I_1\ and\ I_2\ are\ semantically\ dissimilar, \end{cases} \tag{1}$$

where $H(\cdot, \cdot)$ denotes the Hamming distance between two binary codes, and t is a threshold. This formula means that the loss of two similar images is their Hamming distance, the greater Hamming distance of the pairwise images is, the more loss will be produced. Otherwise the Hamming distance of two dissimilar images which is within the threshold can contribute to the loss. S denotes whether the pair of images are similar to each other. If two images are similar, $S = 1$, otherwise $S = 0$.

It would be very preferable to directly use the pairwise loss function to train the network with back-propagation algorithm. However, it is difficult to use Eqn. 1 due to its non-differentiable property. To handle this problem, a commonly used method is to utilize sigmoid or tanh to restrict the outputs to be within $[-1, 1]$, which relaxes the integer constraint into range constraint. But this kind of methods will result in restraining the convergence of the network. So we relax the binary limitation by replacing the Hamming distance with Euclidean distance to get the real-valued outputs of the network, and replace $\{-1, 1\}$ with $[-1, 1]$. Eqn. 1 can be rewritten as:

$$W_1(b_1, b_2) = \begin{cases} \frac{1}{2}\|b_1 - b_2\|^2 & S = 1 \\ \frac{1}{2}max(t - \|b_1 - b_2\|^2, 0) & S = 0 \end{cases}$$
$$s.t.\ b_i \in [-1, +1]^q\ ,\quad i \in \{1, 2\}. \tag{2}$$

The l_2-norm is utilized to measure the distance between the outputs of the network. With the aforementioned pairwise loss function, the network is trained with mini-batch gradient descent method using back-propagation algorithm. The gradient of Eqn. 2 w.r.t. $\mathrm{b}_i, i \in \{1, 2\}$ can be computed as:

$$\frac{\partial W_1}{\partial b_i} = (-1)^{i+1}(b_1 - b_2) \tag{3}$$

$$\frac{\partial W_1}{\partial b_i} = \begin{cases} (-1)^i(b_1 - b_2) & \|b_1 - b_2\|^2 < t \\ 0 & \|b_1 - b_2\|^2 \geq t \end{cases}$$
$$s.t.\ b_i \in [-1, +1]^q\ ,\quad i \in \{1, 2\}, \tag{4}$$

when $S = 1$, the gradient can be computed as Eqn. 3, otherwise $S = 0$, the gradient can be computed as Eqn. 4.

Due to the fact that the outputs of the network are relaxed to be real-valued, the process to map the outputs into binary codes from Euclidean space to Hamming space will produce quantization loss. After obtaining the outputs of the network, the last simple quantization step can be written as:

$$b = sign(v), \tag{5}$$

where v is the outputs of our network and $sign(v)$ is the sign function on the output vectors that $sign(v(i)) = 1$ if $v(i) > 0$, otherwise $sign(v(i)) = -1$, for $i = 1, 2, \ldots, q$.

To overcome the quantization loss from Euclidean space to Hamming space and preserve the information of original data, a quantization loss function is proposed, which can be written as:

$$W_2 = \frac{1}{q}\sum_{i=1}^{q} \|b(i) - v(i)\|^2\ , \tag{6}$$

where $b(i)$ denotes the i-th bit of the binary code, q denotes the length of binary code and $v(i)$ represents one of the network real-valued output units. We calculate the difference between the original outputs $v(i)$ and the result of quantization $b(i)$.

According to the information theory, the higher entropy is, the more information can be carried. We encourage the compact binary codes to be evenly distributed in order to increase the information capacity. When the probability of 1 or -1 in each bit is more close to 50% respectively, the more information can be carried. So the average of all bits is enforced to be close to zero. The loss function can be defined as:

$$W_3 = \frac{1}{q}\sum_{j=1}^{q}\left\|\frac{1}{n}\sum_{i=1}^{n} b_i(j) - 0\right\|^2, \tag{7}$$

where $b_i(j)$ denotes j-th bit of i-th binary code, n denotes the number of binary codes.

With aforementioned loss functions, the proposed method enforce the network to preserve the semantic similarity of pairwise images. At the same time, the loss of quantization and uneven distribution are minimized to obtain high-quality binary codes.

The final loss function can be written as:

$$\begin{aligned} W = {} & \frac{1}{2C_n^2}\sum_{i=1}^{C_n^2}(SH(b_1, b_2) + (1 - S)max(t - H(b_1, b_2), 0)) \\ & + \frac{1}{2n}\sum_{i=1}^{n}\frac{1}{q}\sum_{j=1}^{q}\|b_i(j) - v_i(j)\|^2 \\ & + \frac{1}{2q}\sum_{j=1}^{q}\left\|\frac{1}{n}\sum_{i=1}^{n} b_i(j) - 0\right\|^2, \end{aligned} \tag{8}$$

C. SDHP+

The pairwise images are organized to preserve the original semantic relationship of images as mentioned in Section III-B, which only utilize the similarity relation between the pairwise labels. The learning of the hash functions are not associated with the individual class information. Due to the fact that the discrete class labels of the individual images are also available and the classification power has been demonstrated in [27], an additional idea is come up, which extends the framework to further improve the sematic feature discriminative power of the learned binary codes, and make full use of the supervised information. The framework is defined as SDHP+, which integrates the class label information of each training image into the network. A new fully connected layer is added to the network on the juxtaposition with the hashing layer, which has c units representing c classes, so the top layer of the network has $q + c$ units in total. It can be regarded as a transfer learning case in which the incorporated additional image class labels predicting is expected to be helpful for learning a more accurate image representation such that it may be advantageous for the learning of hashing functions.

Two optimization objectives are employed to the new layer of the deep network. Firstly, $L_1(i, z) = -log(\frac{e^{z_i}}{\sum_{j=1}^{m} e^{z_j}})$, a commonly used softmax loss function for classification is utilized for training with discrete class label information of individual image. Due to the fact that one image may associate with many labels, the label of each training image is represented to a vector $Y \in \{0, 1\}^c$, where c is the number of classes, if the image belongs to i-th class, $Y_i = 1$; otherwise, $Y_i = 0$. Secondly, aforementioned pairwise loss function is also the optimization objective of this layer as defined in Eqn. 2. The similarity of pairwise images can be defined as:

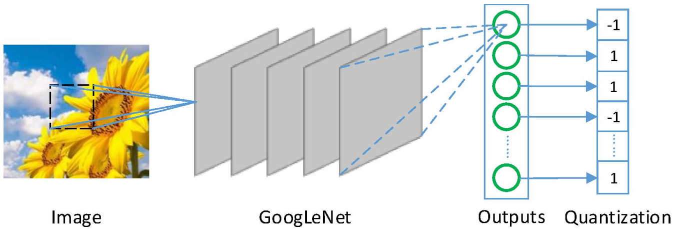


Fig. 2. The architecture of prediction.

$$S = \begin{cases} 1 & label_1 = label_2 \\ 0 & label_1 \neq label_2, \end{cases} \tag{9}$$

$$S = label_1 \ \& \ label_2, \tag{10}$$

If each image is associated with a single label, S can be computed with Eqn. 9, otherwise bitwise AND operators of pairwise labels are executed as Eqn. 10 shown.

With above devised double restrictions, SDHP+ enforces the learned binary codes to achieve better search accuracy.

D. Hash Coding for New Images

After the training process of the network is completed, it can be used to generate a q-bit binary code for a new input image. As shown in Fig. 2, an image is firstly input into the network and encoded into a q-dimensional real-valued feature vector v. We only utilize the q outputs of SDHP, and c outputs of SDHP+ are only used for training. Then a q-bit binary code can be obtained by a simple quantization step $b = sign(v)$ for the outputs of the network as mentioned before.

IV. Experiments and Results

A. Datasets and Evaluation Protocols

The experiments are conducted on two commonly used public benchmark image datasets CIFAR-10 and NUS-WIDE. These two datasets are more closely to natural scenes, including various targets and scenes might appear in the driving environment.

- **CIFAR-10** [45] consists of 60,000 32×32 color images which are categorized into 10 classes, and each class contains 6,000 images. It is a single-label dataset in which each image belongs to one of the ten classes. The dataset is split into training set and test set, with 50,000 and 10,000 images respectively. For conventional hashing methods, 512-D GIST descriptors is utilized to represent images, following [34]. For our method, the raw images are used as the input of the framework.

TABLE I

MEAN AVERAGE PRECISION (MAP) ON CIFAR-10. * REPRESENTS CITED FROM THE ORIGINAL PAPERS

Method	12 bits	24 bits	36 bits	48 bits
SDHP	**0.8318**	**0.8684**	**0.8755**	**0.8767**
LSH	0.1217	0.1218	0.1434	0.1417
PCAH	0.1311	0.1290	0.1255	0.1235
SH	0.1268	0.1242	0.1238	0.1282
ITQ	0.1548	0.1649	0.1668	0.1684
DSH	0.1454	0.1567	0.1589	0.1652
SDH	0.4054	0.5139	0.5347	0.5377
COSDISH	0.4804	0.5118	0.5413	0.5493
CNNH*	0.4650	0.5210	-	0.5320
DNNH*	0.5520	0.5660	-	0.5810
DSH*	0.6157	0.6512	0.6607	0.6755

- **NUS-WIDE** [46] is web image dataset. The dataset includes nearly 270,000 images. It is a multi-label dataset. Each image of the dataset is associated with at least one class label from 81 semantic concepts. We follow the setting in [47] to use the images associated with the 21 most frequent labels, and each label includes at least 5,000 images. The total of the images is 195,834. 10,000 images are randomly chosen as test set, and the rest are used as training set. For conventional methods, the provided 225-D block-wise color moments low-level feature is utilized as the input.

In the experiments, for CIFAR-10, if two images have the same label, they are considered to be semantically similar to each other, and vice versa. For NUS-WIDE, due to the multi-label property, it is defined that if two images share at least one label, they are considered semantically similar, and otherwise they are dissimilar.

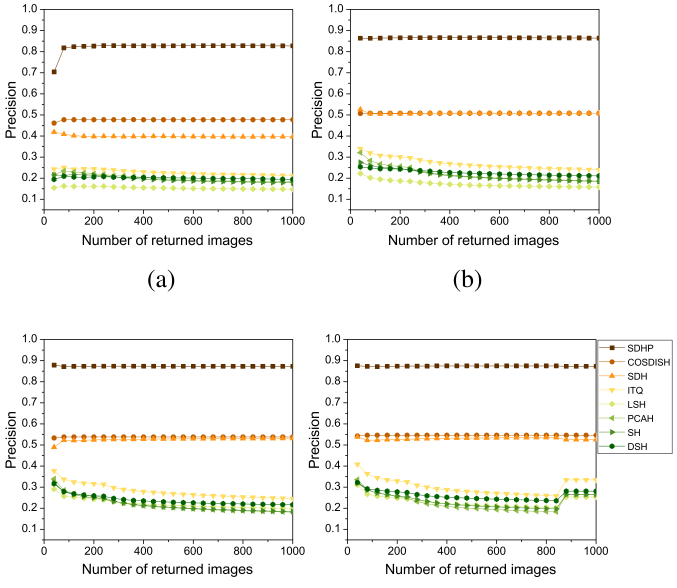
The proposed method is implemented on Caffe [48] framework with a single Tesla K20c GPU (5GB memory), and compared with several state-of-the-art conventional algorithms, include LSH, PCAH, SH, ITQ, DSH, COSDISH, SDH. The results of these baseline methods are obtained by the open-source implementation provided by their authors. We also compare the method with several deep hashing algorithms, include CNNH, DNNH and DSH. Since the implementation of these methods are not available and our experimental setup is similar to them for the datasets, we use the numbers reported in the original paper for reference [34], [35], [39].

The retrieval quality are evaluated based on four evaluation metrics: (1) the Mean Average Precision (MAP), (2) precision of the top 1,000 returned images using Hamming ranking, (3) precision-recall curve using Hamming ranking, (4) precision within Hamming radius 2.

In all experiments, the network is trained by stochastic gradient descent with 0.9 momentum, the mini-batch size of images is 50 and the weight decay parameter is 0.004.

B. Results on CIFAR-10

Table I lists the Mean Average Precision with different binary code lengths. As shown in Table I, we can observe that the proposed method achieves better search accuracy than the baseline algorithms. For example, compared to the second best competitor DSH* which also uses pairwise images as inputs, the MAP results of SDHP increase 20.12% ∼ 21.72%. Table II presents the precision of the top 1,000 returned images using Hamming ranking on CIFAR-10. The precisions of the top 1,000 returned images using Hamming ranking are above 82%, 86%, 87%, 87% with 12 bits, 24 bits, 36 bits, 48 bits respectively. The results demonstrate that SDHP has better performance even with short code lengths, and the precision is far beyond the conventional algorithms.



(a) (b)

(c) (d)

Fig. 3. Precision curve with regard to top-*n* with different bits on CIFAR-10. (a) 12 bits. (b) 24 bits. (c) 36 bits. (d) 48 bits.

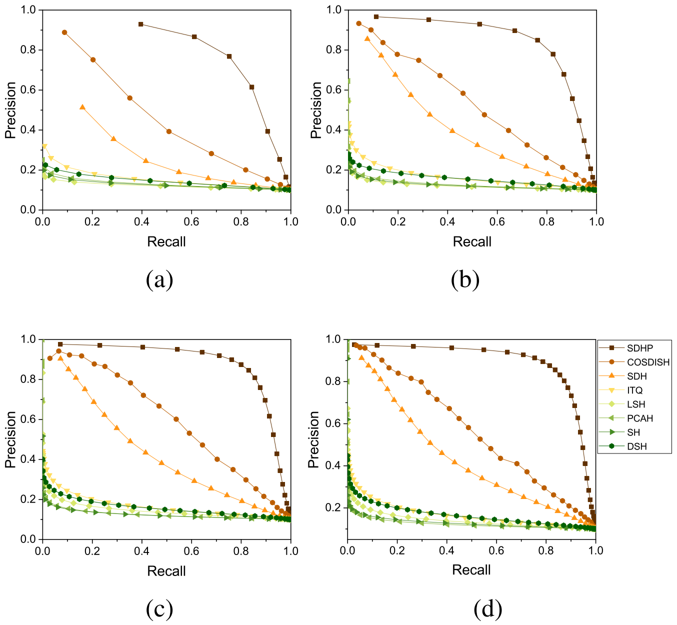
TABLE II

PRECISION OF THE TOP 1,000 RETURNED IMAGES USING HAMMING RANKING ON CIFAR-10

Method	12 bits	24 bits	36 bits	48 bits
SDHP	**0.8272**	**0.8645**	**0.8724**	**0.8748**
LSH	0.1480	0.1579	0.1992	0.1994
PCAH	0.1833	0.1863	0.1818	0.1784
SH	0.1784	0.1852	0.1842	0.1925
ITQ	0.2123	0.2400	0.2458	0.2523
DSH	0.1946	0.2110	0.2165	0.2317
SDH	0.3967	0.5079	0.5311	0.5363
COSDISH	0.4775	0.5082	0.5382	0.5463

As Fig. 3 exhibits, the precision with different code lengths with regard to different number of top returned samples using Hamming ranking are above 80% approximately, and the precision with 48 bits are more than 87%. Compared to the best conventional hashing method, the precision increases by about 30%. As fig. 4 displays, SDHP generally outperforms all comparison methods by large margins in the metrics of precision-recall curves using Hamming ranking with different bits.

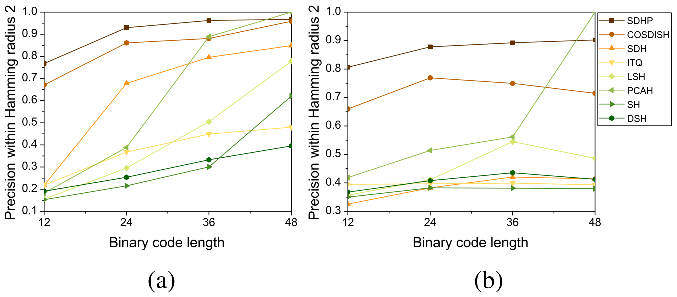
Fig. 5 (a) shows the precision within Hamming radius 2 on CIFAR-10. The retrieval performance using Hamming ranking within Hamming radius 2 is important for retrieval with binary codes, because such Hamming ranking only requires constant time cost. As shown in the figure, our method achieves higher



(a) (b)

(c) (d)

Fig. 4. Precision-recall curves of Hamming ranking with regard to different number of bits on CIFAR-10. (a) 12 bits. (b) 24 bits. (c) 36 bits. (d) 48 bits.



(a) (b)

Fig. 5. Precision within Hamming radius 2 on (a) CIFAR-10 and (b) NUS-WIDE.

TABLE III

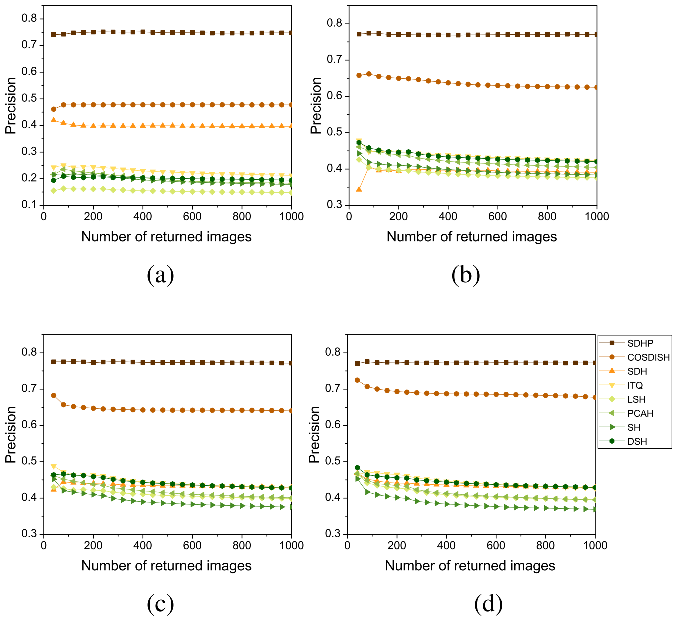
MEAN AVERAGE PRECISION (MAP) ON NUS-WIDE. * REPRESENTS CITED FROM THE ORIGINAL PAPERS

Method	12 bits	24 bits	36 bits	48 bits
SDHP	**0.7507**	**0.7720**	**0.7758**	**0.7748**
LSH	0.3326	0.3386	0.3552	0.3469
PCAH	0.3523	0.3448	0.3412	0.3386
SH	0.3468	0.3447	0.3400	0.3374
ITQ	0.3525	0.3560	0.3575	0.3580
DSH	0.3461	0.3543	0.3543	0.3511
SDH	0.4185	0.4002	0.4396	0.4405
COSDISH	0.6398	0.6575	0.6772	0.7136
CNNH*	0.6230	0.6300	-	0.6250
DNNH*	0.6740	0.6970	-	0.7150
DSH*	0.5483	0.5513	0.5582	0.5621

precision than other algorithms with 12 bits, 24 bits, 36 bits, and the precisions are over 90% with 24 bits, 36 bits, 48 bits.

C. Results on NUS-WIDE

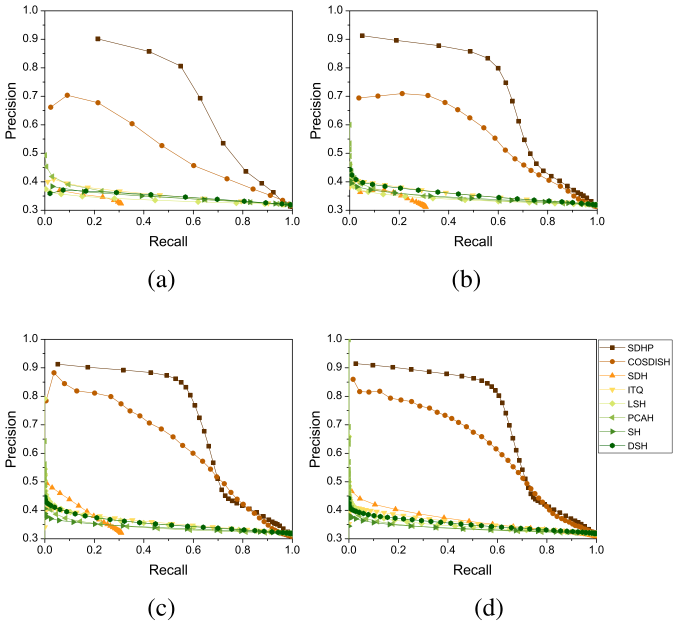
Table III lists the Mean Average Precision with different binary code lengths on NUS-WIDE. It is obviously that the performance of SDHP is also better than other approaches on NUS-WIDE. The MAP of the proposed method can reach about 77.58% with 36 bits. Compared to the second best competitor DNNH*, the MAP increase 5.98% ∼ 7.67%. The MAP results illustrate that our method can achieve better search accuracy than the baseline algorithms as well as CIFAR-10. Table IV shows the precision of the top 1,000 returned images using Hamming ranking are above 74% with 12 bits and 77% with 24 bits, 36 bits, 48 bits respectively on NUS-WIDE.



(a) (b)

(c) (d)

Fig. 6. Precision curves with regard to top-n with different bits on NUS-WIDE. (a) 12 bits. (b) 24 bits. (c) 36 bits. (d) 48 bits.



(a) (b)

(c) (d)

Fig. 7. Precision-recall curves of Hamming ranking with regard to different number of bits on NUS-WIDE. (a) 12 bits. (b) 24 bits. (c) 36 bits. (d) 48 bits.

Fig. 6 presents that the precision with regard to different number of top returned samples are approximately 77%. Compared to the best conventional hashing method, the precision increases by approximately 14%. Fig. 5 (b) exhibits the precision within Hamming radius 2 on NUS-WIDE.

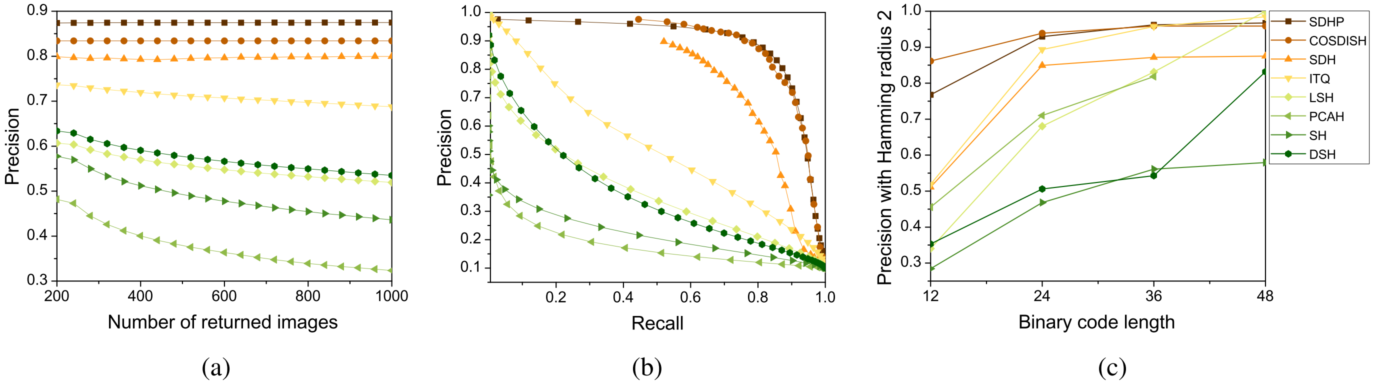


Fig. 8. The results of comparison methods on CNN features of CIFAR-10: (a) Precision curves with 48 bits w.r.t. different number of top returned samples, (b) Precision-recall curves of Hamming ranking with 48 bits and (c) Precision within Hamming radius 2.

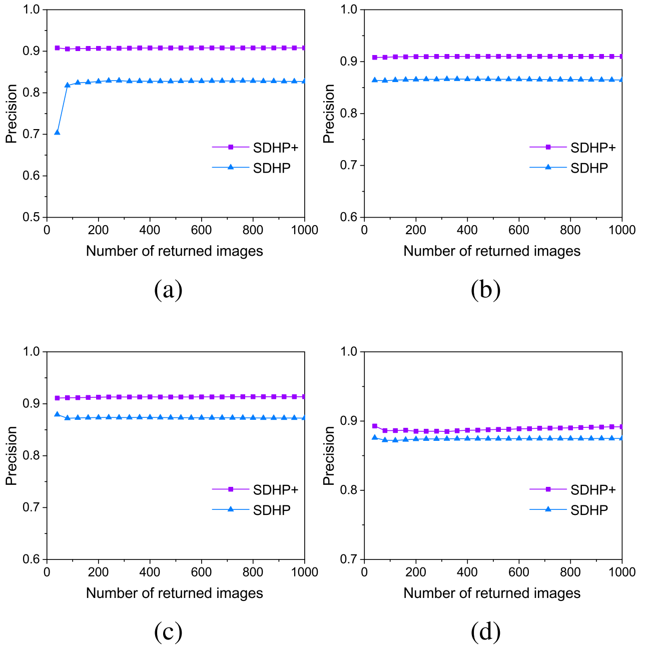


Fig. 9. Precision curves with regard to top-*n* with different bits on CIFAR-10. (a) 12 bits. (b) 24 bits. (c) 36 bits. (d) 48 bits.

TABLE IV
PRECISION OF THE TOP 1,000 RETURNED IMAGES USING HAMMING RANKING ON NUS-WIDE

Method	12 bits	24 bits	36 bits	48 bits
SDHP	**0.7477**	**0.7705**	**0.7715**	**0.7723**
LSH	0.3497	0.3760	0.3986	0.3957
PCAH	0.4118	0.4047	0.4008	0.3949
SH	0.3788	0.3836	0.3754	0.3689
ITQ	0.4193	0.4227	0.4277	0.4274
DSH	0.4106	0.4205	0.4276	0.4292
SDH	0.4073	0.3907	0.4299	0.4292
COSDISH	0.6000	0.6250	0.6401	0.6773

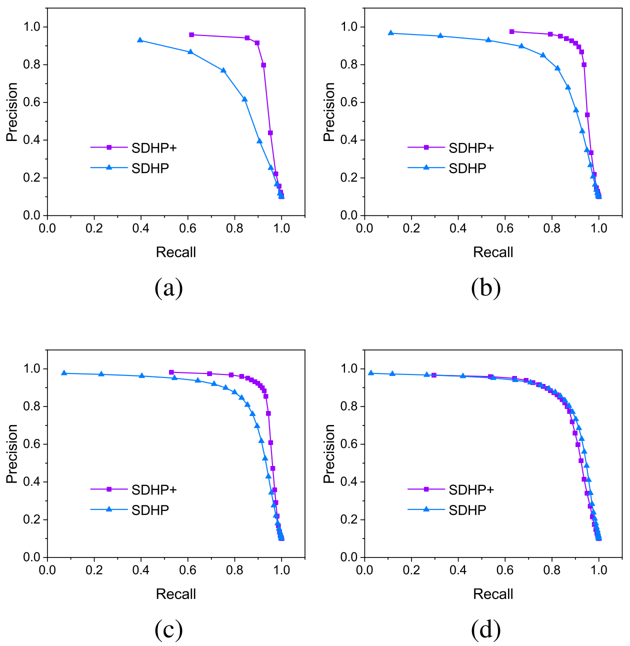


Fig. 10. Precision-recall curves of Hamming ranking with regard to different number of bits on CIFAR-10. (a) 12 bits. (b) 24 bits. (c) 36 bits. (d) 48 bits.

The proposed method achieves higher precision than others with 12 bits, 24 bits, 36 bits, and the precisions are approximately 90%. As fig. 7 shows the precision-recall curves using Hamming ranking with different code lengths, it can be seen that SDHP also achieves better search accuracy and outperforms all baseline methods by large margins on NUS-WIDE.

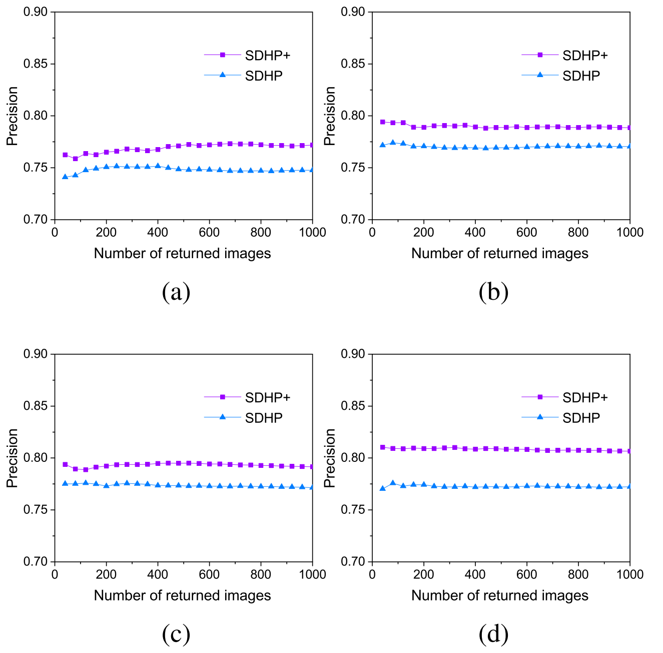
The substantial superior performance demonstrates that the deeper architecture can learn a good image representation as well as hash functions and the proposed loss functions can encourage to obtain the high-quality binary codes for image retrieval.

D. Compare With Conventional Hashing Methods Using CNN Features

In general, the hashing method based on deep learning outperforms the conventional hashing algorithms with hand-crafted features. In order to verify the performance of the learned hashing functions by SDHP, the experimental comparisons are conducted between SDHP and conventional hashing algorithms regardless of the effect of CNN features. These approaches are trained with CNN features instead of hand-crafted features. We extract the features from the original

TABLE V
MEAN AVERAGE PRECISION (MAP) ON CNN FEATURES OF CIFAR-10

Method	12 bits	24 bits	36 bits	48 bits
SDHP	**0.8318**	**0.8684**	**0.8755**	**0.8767**
LSH	0.2300	0.2868	0.3412	0.3667
PCAH	0.2773	0.2174	0.1926	0.1795
SH	0.2514	0.2413	0.2293	0.2196
ITQ	0.4955	0.5177	0.5213	0.5289
DSH	0.3420	0.3056	0.3306	0.3561
SDH	0.7711	0.7964	0.8011	0.8021
COSDISH	0.8037	0.8155	0.8270	0.8304



(a) (b)

(c) (d)

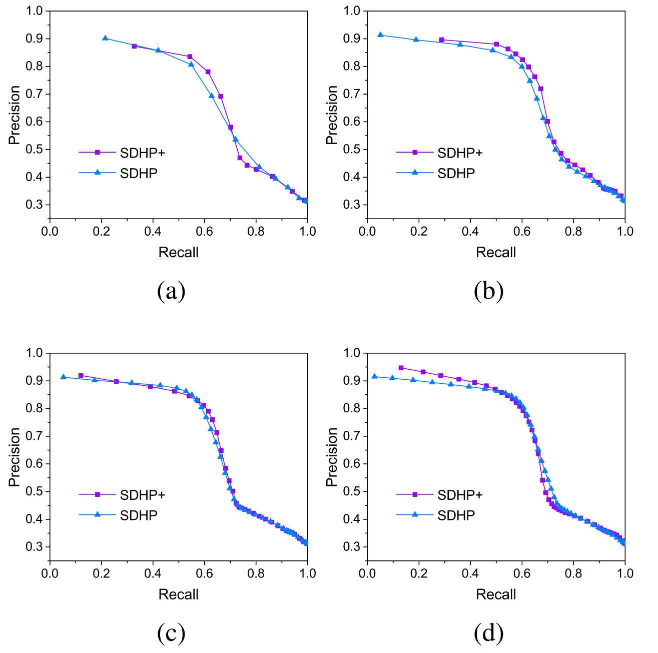
Fig. 11. Precision curves with regard to top-n with different bits on NUS-WIDE. (a) 12 bits. (b) 24 bits. (c) 36 bits. (d) 48 bits.

outputs of the fine-tuned GoogLeNet on CIFAR-10, which are 1,000-dimensional feature vectors. The results of all hashing methods with CNN features are listed in table V and fig. 8. As listed in table V, the MAP of conventional hashing methods achieve better search accuracy than the results on hand-crafted features, due to the increased dimension of features and the learning power of CNN. However, as we can see, SDHP still outperforms conventional hashing algorithms, the MAP exceeds the second competitor approximately 5%.

Fig. 8 shows (a) precision curves with regard to different number of top returned samples using 48-bit binary codes, (b) precision-recall curves of Hamming ranking with 48 bits and (c) precision within Hamming radius 2. It demonstrates that our method performs well in spite of the utilization of CNN features to conventional hashing algorithms. Although the performance of the baseline methods with CNN features improve a lot, it is obviously that the total time cost of both feature extraction and hashing quantization increase as well.

E. Results on SDHP+

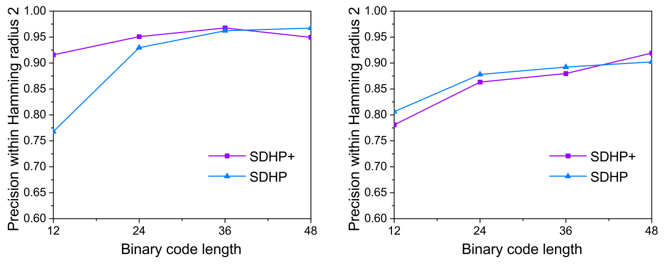
In order to verify the performance of SDHP+, the experimental comparisons are conducted between SDHP+ and SDHP. The evaluation metrics are the same as above experiments. Table VI lists the MAP of SDHP+ and SDHP with different code lengths, which presents the map of SDHP+ exceeding SDHP with 12 bits more than 7.9%.



(a) (b)

(c) (d)

Fig. 12. Precision-recall curves of Hamming ranking with regard to different number of bits on NUS-WIDE. (a) 12 bits. (b) 24 bits. (c) 36 bits. (d) 48 bits.



(a) (b)

Fig. 13. Precision within Hamming radius 2 on (a) CIFAR-10 and (b) NUS-WIDE.

TABLE VI
MEAN AVERAGE PRECISION (MAP) ON CIFAR-10

Method	12 bits	24 bits	36 bits	48 bits
SDHP+	**0.9116**	**0.9151**	**0.9178**	**0.9002**
SDHP	0.8318	0.8684	0.8755	0.8767

Fig. 9 shows the precision curves with regard to top-n on CIFAR-10, it can be seen that the precision curve of SDHP+ is above SDHP by a large margin. As listed in table VII, the precision of the top 1,000 returned images exceeds SDHP more than 8% with 12 bits. Fig. 10 presents the precision-recall curves of Hamming ranking with different code lengths. SDHP+ also outperforms SDHP by relatively large margins using Hamming ranking with 12, 24, 36 bits.

The same comparisons are also conducted on NUS-WIDE. As listed in table VIII, the MAP of SDHP+ exceeds SDHP 2.05% with 12 bits, and the MAP of SDHP+ is above 81% with 48 bits. Fig. 11 presents the precision curves with regard to top-n with different bits on NUS-WIDE, it is

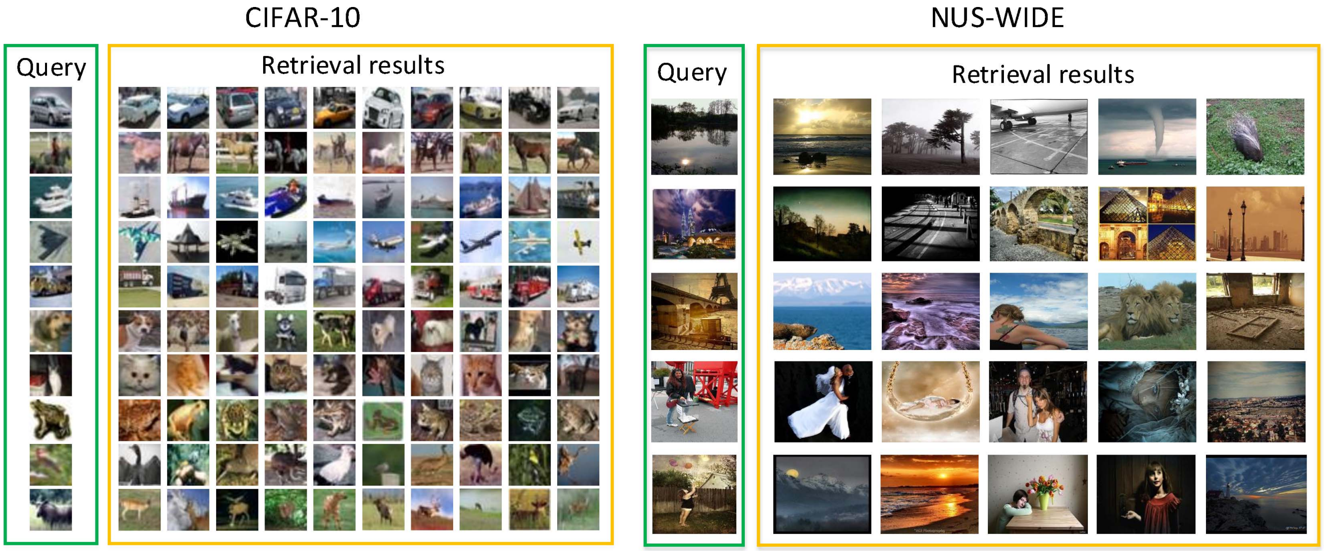


Fig. 14. Retrieval results on CIFAR-10 (left) and NUS-WIDE (right). Ten images are randomly selected from the returned set for ten CIFAR-10 test images using Hamming ranking on 12-bit hash codes and five images are randomly selected from the returned set for five NUS-WIDE test images using Hamming ranking on 48-bit hash codes.

TABLE VII

PRECISION OF THE TOP 1,000 RETURNED IMAGES USING HAMMING RANKING ON CIFAR-10

Method	12 bits	24 bits	36 bits	48 bits
SDHP+	**0.9080**	**0.9100**	**0.9135**	**0.8918**
SDHP	0.8272	0.8645	0.8724	0.8748

TABLE VIII

MEAN AVERAGE PRECISION (MAP) ON NUS-WIDE

Method	12 bits	24 bits	36 bits	48 bits
SDHP+	**0.7712**	**0.7919**	**0.7953**	**0.8108**
SDHP	0.7507	0.7720	0.7758	0.7748

TABLE IX

PRECISION OF THE TOP 1,000 RETURNED IMAGES USING HAMMING RANKING ON NUS-WIDE

Method	12 bits	24 bits	36 bits	48 bits
SDHP+	**0.7719**	**0.7886**	**0.7915**	**0.8066**
SDHP	0.7477	0.7705	0.7715	0.7723

obvious that the precision of SDHP+ is higher than SDHP. Table IX presents the precision of the top 1,000 returned images exceeds SDHP approximately 2% with 12, 24, 36 bits. As fig. 12 shows, the precision-recall curves of Hamming ranking with regard to different code lengths of SDHP+ are close to SDHP, but it still can be seen that SDHP+ is slightly superior to SDHP. Fig. 13 (a) (b) displays the precision within Hamming radius 2 on CIFAR-10 and NUS-WIDE respectively, the precision reaching to 96% with 36 bits on CIFAR-10 and 91% with 48 bits on NUS-WIDE.

It demonstrates that the idea of SDHP+ is effective, and it is obvious that the performance of SDHP+ is better than SDHP even with less bits.

Fig. 14 presents the retrieval results on CIFAR-10 and NUS-WIDE. For CIFAR-10, ten query images are randomly selected from the test set, and 12-bit binary codes are extracted for retrieval. The images which have the same binary codes with the query are added into the returned set, and then ten images from the returned set of each query are randomly chose to display. For NUS-WIDE, we choose five test images as the query images with 48-bit binary codes, and five returned images of each query are randomly selected. It is verified that our method can achieve satisfactory retrieval performance, and it is supposed to be effective for intelligent vehicles to recognize the environment.

TABLE X

ENCODING TIME (IN SECOND) OF DIFFERENT HASHING METHODS ON THE CIFAR-10 DATASET USING 48 BITS

Method	Encoding time
LSH	2.82e-6
PCAH	3.28e-6
SH	2.37e-5
ITQ	7.99e-6
DSH	2.68e-6
SDH	2.17e-2
COSDISH	1.87e-3
SDHP+	9.83e-3
CNN features	9.89e-3

F. Computational Time

Table X shows the encoding time of different hashing methods on the CIFAR-10 dataset using 48 bits. The extracting time of CNN features is also listed. It can be seen that the encoding time of SDHP+ is closely to the feature extraction time of GoogLeNet, and the encoding time of our method is lower than SDH. The proposed method can achieve significantly better performances compared to these methods while being scalable with any large-scale training data thanks to the batch process manner.

G. Evenly Distribution of the Binary Codes

To verify if the distribution of the learned binary codes are evenly, the experiments are conducted. We count the number

TABLE XI
EVENLY DISTRIBUTION OF THE BINARY CODES ON CIFAR-10

12 bits	24 bits	36 bits	48 bits
1.0405	1.2667	1.2619	1.2380

of -1 and 1 in a same bin of all binary codes respectively, and calculate the ratio of them. The results on CIFAR-10 are shown in table XI.

The results shows the ratio of -1 and 1 is close to 1, which demonstrates that the loss function is effective to make the binary codes distribute evenly.

V. CONCLUSION

In this paper, a simple yet effective supervised one-stage deep hashing framework is elaborated designed to obtain more discriminative binary codes and achieve promising retrieval performance for intelligent vehicles to recognize the environment. The contributions of the proposed method mainly focus on four aspects: first, we choose a deeper network as the basic structure due to its impressive feature representation power and the original network structure is changed to adapt to hash task. Second, pairwise supervised information is utilized and a pairwise loss function is devised to preserve the semantic similarities of the original data meanwhile. Third, the quantization error from Euclidean space to Hamming space is minimized and the binary codes are enforced to be evenly distributed to carry more information. Fourth, the performance is further improved by integrating more supervised information. Extensive experimental results on two benchmark databases demonstrate that our method outperforms many state-of-the-art algorithms, and it is supposed to be effective for environment perception of intelligent vehicles.

REFERENCES

[1] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1773–1795, Dec. 2013.

[2] L. Nie, M. Wang, Z.-J. Zha, and T.-S. Chua, "Oracle in image search: A content-based approach to performance prediction," *ACM Trans. Inf. Syst.*, vol. 30, no. 2, pp. 13-1–13-23, 2012.

[3] H. Xie, Y. Zhang, J. Tan, L. Guo, and J. Li, "Contextual query expansion for image retrieval," *IEEE Trans. Multimedia*, vol. 16, no. 4, pp. 1104–1114, Jun. 2014.

[4] H. Jegou, M. Douze, and C. Schmid, *Hamming Embedding Weak Geometric Consistency for Large Scale Image Search*. Berlin, Germany: Springer, 2008.

[5] J. Wang, Y. Song, T. Leung, and C. Rosenberg, "Learning fine-grained image similarity with deep ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1386–1393.

[6] J. Wang, H. T. Shen, J. Song, and J. Ji, "Hashing for similarity search: A survey," *CoRR*, vol. abs/1408.2927, 2014.

[7] Q. Wang, G. Zhu, and Y. Yuan, "Statistical quantization for similarity search," *Comput. Vis. Image Understand.*, vol. 124, pp. 22–30, Jul. 2014.

[8] J. Fang, H. Xu, Q. Wang, and T. Wu, "Online hash tracking with spatio-temporal saliency auxiliary," *Comput. Vis. Image Understand.*, vol. 160, pp. 57–72, Jul. 2017.

[9] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proc. Int. Conf. Very Large Data Bases*, vol. 8. 1999, pp. 518–529.

[10] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," in *Proc. 47th Annu. IEEE Symp. Found. Comput. Sci. (FOCS)*, Oct. 2006, pp. 459–468.

[11] O. Chum *et al.*, "Near duplicate image detection: Min-Hash and *tf-idf* weighting," in *Proc. BMVC*, vol. 810. 2008, pp. 812–815.

[12] P. Jain, B. Kulis, and K. Grauman, "Fast image search for learned metrics," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.

[13] M. Raginsky and S. Lazebnik, "Locality-sensitive binary codes from shift-invariant kernels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1509–1517.

[14] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1753–1760.

[15] J. Wang, S. Kumar, and S.-F. Chang, "Sequential projection learning for hashing with compact codes," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 1127–1134.

[16] Y. Gong and S. Lazebnik, "Iterative quantization: A procrustean approach to learning binary codes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 817–824.

[17] F. Shen, C. Shen, W. Liu, and H. Tao Shen, "Supervised discrete hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 37–45.

[18] W.-C. Kang, W.-J. Li, and Z.-H. Zhou, "Column sampling based discrete supervised hashing," in *Proc. 13th AAAI Conf. Artif. Intell.*, Feb. 2016, pp. 1230–1236.

[19] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.

[20] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1. Jun. 2005, pp. 886–893.

[21] H. Xie *et al.*, "Robust common visual pattern discovery using graph matching," *J. Vis. Commun. Image Represent.*, vol. 24, no. 5, pp. 635–646, Jul. 2013.

[22] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 12, pp. 1349–1380, Dec. 2000.

[23] L. Nie, M. Wang, L. Zhang, S. Yan, B. Zhang, and T. S. Chua, "Disease inference from health-related questions via sparse deep learning," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 8, pp. 2107–2119, Aug. 2015.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[25] M. Lin, Q. Chen, and S. Yan, "Network in network," *CoRR*, vol. abs/1312.4400, 2013.

[26] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[27] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1026–1034.

[29] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2553–2561.

[30] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1988–1996.

[31] C. Yan, H. Xie, S. Liu, J. Yin, Y. Zhang, and Q. Dai, "Effective uyghur language text detection in complex background images for traffic prompt identification," *IEEE Trans. Intell. Transp. Syst.*, to be published.

[32] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3431–3440.

[33] L. Nie, S. Yan, M. Wang, R. Hong, and T.-S. Chua, "Harvesting visual concepts for image search with complex queries," in *Proc. 20th ACM Multimedia Conf. (MM)*, 2012, pp. 59–68.

[34] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, "Supervised hashing for image retrieval via image representation learning," in *Proc. AAAI*, vol. 1. 2014, pp. 2156–2162.

[35] H. Lai, Y. Pan, Y. Liu, and S. Yan, "Simultaneous feature learning and hash coding with deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3270–3278.

[36] V. E. Liong, J. Lu, G. Wang, P. Moulin, and J. Zhou, "Deep hashing for compact binary codes learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2475–2483.

[37] R. Zhang, L. Lin, R. Zhang, W. Zuo, and L. Zhang, "Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 4766–4779, Dec. 2015.

[38] F. Zhao, Y. Huang, L. Wang, and T. Tan, "Deep semantic ranking based hashing for multi-label image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1556–1564.

[39] H. Liu, R. Wang, S. Shan, and X. Chen, "Deep supervised hashing for fast image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2064–2072.

[40] W.-J. Li, S. Wang, and W.-C. Kang. (2015). "Feature learning based deep supervised hashing with pairwise labels." [Online]. Available: https://arxiv.org/abs/1511.03855

[41] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang, "Supervised hashing with kernels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 2074–2081.

[42] G. Lin, C. Shen, Q. Shi, A. van den Hengel, and D. Suter, "Fast supervised hashing with decision trees for high-dimensional data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1963–1970.

[43] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *ICLR*, 2015.

[44] R. Salakhutdinov and G. Hinton, "Semantic hashing," *Int. J. Approx. Reason.*, vol. 50, no. 7, pp. 969–978, Jul. 2009.

[45] A. Krizhevsky, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009.

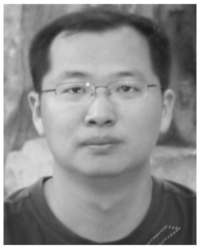
[46] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "NUS-WIDE: A real-world Web image database from national University of Singapore," in *Proc. ACM Int. Conf. Image Video Retr.*, 2009, p. 48.

[47] W. Liu, J. Wang, S. Kumar, and S.-F. Chang, "Hashing with graphs," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 1–8.

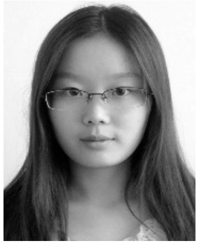
[48] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 675–678.

Chenggang Yan received the B.S. degree in computer science from Shandong University in 2008, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, in 2013.

He is currently a Professor with Hanzhou Dianzi University. Prior to that, he was an Assistant Research Fellow with Tsinghua University. He has authored or co-authored over 30 refereed journal and conference papers. His research interests include machine learning, image processing, computational biology, and computational photography. As a co-author, he got the best paper awards at the International Conference on Game Theory for Networks 2014 and the SPIE/COS Photonics Asia Conference 9273 2014, and the Best Paper Candidate at the International Conference on Multimedia and Expo 2011.



Hongtao Xie received the Ph.D. degree in computer application technology from the Institute of Computing Technology, Chinese Academy of Sciences, China, in 2012. He is currently an Associate Professor with the Institute of Information Engineering, Chinese Academy of Sciences. His research interests include multimedia content analysis and retrieval, similarity search, and parallel computing.



Dongbao Yang received the B.E. degree in computer science and technology from Shandong University, Weihai, in 2015, where she is currently pursuing the master's degree. Her research interests include multimedia content analysis and retrieval, similarity search.

Jian Yin received the Ph.D. degree from Shandong University, Weihai. He is currently an Associate Professor with Shandong University. His research interests include computer software and theory, computer graphics.

Yongdong Zhang (M'08–SM'13) received the Ph.D. degree in electronic engineering from Tianjin University, Tianjin, China, in 2002. He is currently a Professor with the School of Information Science and Technology, University of Science and Technology of China, Hefei, China. His current research interests are in the fields of multimedia content analysis and understanding, multimedia content security, video encoding, and streaming media technology.

He has authored over 100 refereed journal and conference papers. He was a recipient of the best paper awards in PCM 2013, ICIMCS 2013, and ICME 2010, the Best Paper Candidate in ICME 2011. He serves as an Editorial Board Member of *Multimedia Systems Journal* and *Neurocomputing*.

Qionghai Dai (SM'05) received the B.S. degree in mathematics from Shanxi Normal University, Xi'an, China, in 1987, and the M.E. and Ph.D. degrees in computer science and automation from Northeastern University, Shenyang, China, in 1994 and 1996, respectively.

He has been a Faculty Member with Tsinghua University, Beijing, China, since 1997. He is currently a Cheung Kong Professor with Tsinghua University and is the Director of the Broadband Networks and Digital Media Laboratory. His current research interests include signal processing and computer vision and graphics.