IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 23, 2019, accepted August 6, 2019, date of publication September 5, 2019, date of current version September 19, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2939650

Angular Deep Supervised Hashing for Image Retrieval

CHANG ZHOU[1], LAI-MAN PO[1], (Senior Member, IEEE), WILSON Y. F. YUEN[2], KWOK WAI CHEUNG[3], (Member, IEEE), XUYUAN XU[4], KIN WAI LAU[1], YUZHI ZHAO[1], MENGYANG LIU[1], AND PETER H. W. WONG[2], (Senior Member, IEEE)

[1]Department of Electrical Engineering, City University of Hong Kong, Hong Kong
[2]TFI Digital Media Ltd., Hong Kong
[3]Hang Seng University of Hong Kong, Hong Kong
[4]Tencent Video, Tencent Holdings Limited, Shenzhen 518057, China

Corresponding author: Chang Zhou (chanzhou3-c@my.cityu.edu.hk)

This work was supported by the Innovation and Technology Fund (ITF) of Hong Kong Government, City University of Hong Kong, under Grant 9440172.

ABSTRACT Deep learning based image hashing methods learn hash codes by using powerful feature extractors and nonlinear transformations to achieve highly efficient image retrieval. For most end-to-end deep hashing methods, the supervised learning process relies on pair-wise or triplet-wise information to provide an internal relationship of similarity data. However, the use of pair-wise and triplet loss function is limited not only by expensive training costs but also by quantization errors. In this paper, we propose a novel semantic learning based hashing method for image retrieval to optimize the deep features structure in the hash space from a perspective of angular view. Specifically, we proposed an angular hashing loss function that explicitly improve intra-class compactness and inter-class separability between features in hash space. Geometrically, angular hashing loss can be regarded as imposing hash constraints on hypersphere manifold. In order to solve the training problem on the multi-label case, we further designed a dynamic Softmax training strategy that can directly train the network using gradient descent method. Extensive experiments on two well-known datasets of CIFAR-10 and NUS-WIDE demonstrate that the proposed Angular Deep Supervised Hashing (ADSH) method can generate high-quality and compact binary codes, which can achieve state-of-the-art performance as compared with conventional image hashing and deep learning-based hashing methods.

INDEX TERMS Image retrieval, quantization, supervised learning-based hashing, Softmax loss, A-Softmax, neural network, convolutional neural network.

I. INTRODUCTION

With the rapid development of social media and smartphones, huge amount of image data is uploaded to the Internet every minute, such as human face and online products. Most recent researches in visual search use content-based image retrieval (CBIR) [1] without relying on label and text information. Basically, CBIR retrieves images similar to a given query image in terms of visual or semantic similarity. A common CBIR method is to represent database images and query images by handcrafted real-valued features such as SIFT [2] and HOG [3]. Then, the image search can be performed by sorting the images of the database according to the feature distance between each database image and the query image. The image with the smallest distance is considered as the most similar image. In general, these handcrafted features are distinctive with low mismatch probability and good for indexing. However, the high dimensionality of the feature domain makes searching very challenging, especially for large-scale image databases. To address this problem, many hash-based Approximate Nearest Neighbor (ANN) search methods [4]–[7] have been proposed. Since the hash-based approach [8] can encode an image into a compact binary code with similarity preservation, the burden of computation and memory requirements can be reduced.

Hash-based methods can be roughly divided into two categories: data-independent and data-dependent. For data-independent methods, the hash function is generated

The associate editor coordinating the review of this manuscript and approving it for publication was Jeon Gwanggil.

independently without training data, such as Local Sensitive Hash (LSH) [9], which randomly projects the image into the feature space and then performs binarization for generating the binary code. For data-dependent methods, hash functions attempt to learn from training data, commonly referred to as Learning to Hash (L2H) algorithms, including Iterative Quantization (ITQ) [10], K-means Hashing (KMH) [11], and Minimum Loss Hash (MLH) [12].

Recently, deep neural networks have demonstrated the effectiveness of end-to-end representation learning and hash coding using nonlinear hash functions. With deep neural network as the bedrock, many deep learning L2H algorithms are proposed. For example, Xia *et al.* [13] combines Convolutional Neural Network (CNN) with the hash function. They proposed CNNH [13], a two-stage training method that learns good image rendering and a set of hash functions. Later, the connections between deep hashing and metric learning began to be noticed, and the latter were widely used for visual object recognition and verification tasks [14]–[16]. The core idea of metric learning is to learn a feature space that brings similar images closer to each other while making dissimilar images farther apart. Inspired by this idea, DSH [17] is proposed with supervision under contrastive loss function using pair-wise training samples. Similarly, DPSH [18] uses the likelihood of image pairs to simultaneously perform feature learning and hash code learning. In addition, DTSH [19] and DNNH [20] use triplet loss training, and triplets are assumed to contain more information than image pairs. To further improve performance, much work has been done to solve the quantization error problem. For example, HashNet [21] propose continuous activation function to address ill-posed gradients. DQN [22] controls the quantization error by using a product quantized codebook, and DHN [23] imposes a pair-wise quantization loss to constrain the output around -1 and 1.

Metric learning, which is closely related to deep hash learning, can achieve additional intra-class compactness and inter-class separability in the hash space. But it has the disadvantage of preventing the network from learning high-quality hash codes. First, metric learning based methods typically utilize pair-wise or triplet-wise labels as supervised information, requiring complex sample selection strategies to identify hard samples during training. This complicates the training process with the number of training pairs or triplets reaching $O(N^2)$ or $O(N^3)$. Second, in order to solve the non-smooth discrete optimization problem, many methods use relaxed binary constraints to control the performance degradation of quantization. However, it is difficult to completely avoid the uncontrollable quantization errors caused by binarizing the continuous embedding into hash codes.

Geometrically, in the binarization process based on *sign* function, the metric learning-based hashing methods usually follow region-to-point pattern when implementing binarization. For example, contrastive loss encourages relative similarity relation among embeddings by forming region based feature space as shown on the left side of Figure 1. In the process, the same color data points should be pulled together in same region, and different color data points should be pushed away. The hashing method can also be considered as an angle-based approach with each hash code in the hash space representing by a range of angular spaces, as shown on the right side of Figure 1. This angular margin-based hashing approach may help achieve better hash code generation. Of course, the solution to the angular margin of the hashing method is to use Softmax loss, which allows the Convolutional Neural Network (CNN) to learn angular features. Recently, deep face recognition studies have also validated the effectiveness of angular margin on continuous space. For example, SphereFace [24] constructs a discriminant of hyperspherical manifolds, and NormFace [25] is interpreted to constrain learning features with L2 norms.

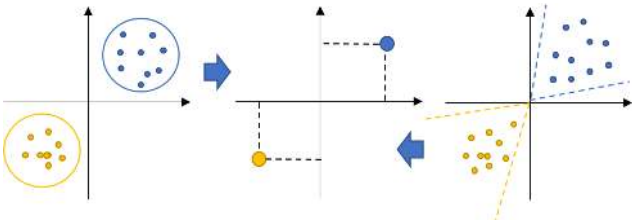


FIGURE 1. Left: Region-to-point hashing pattern, and Right: Angle-to-point hashing pattern. Different color points belong to different categories.

On the other hand, there are very few attempts to implement hashing methods by preserving semantic information. DCWH [26] uses a normalized probability model to learn compact feature spaces, while SSDH [27] trains network to minimize objective function defined on classification errors and other expected hash code attributes. Similar to this work, DLBHC [28] uses Softmax loss to supervise the hidden layers to represent potential concepts. However, the conventional Softmax loss cannot generate compact feature and may result in suboptimal hash code. Therefore, we ask a question whether we can generate compact hash features under the Softmax loss framework, which has angular margin and efficient training. To answer this question, we proposed Angular Deep Supervised Hashing (ADSH) method that can generate centralized and compact hash codes. Basically, we use A-Softmax [24] to encourage intra-class compactness and inter-class differences. Then, we further consider the relative structure of deep features in the hash space. Specifically, we propose a novel angular hashing loss to guide the direction of the angular weights to maximum feature separability in the hash space.

Overall, the proposed method has three desirable advantages.

(1) The proposed angular hashing method has a clear geometric interpretation and is supervised by a flexible learning objective with adjustable constraints.
(2) The deep features learned by the proposed method can naturally adapt to the hash space, which can generate high-quality binary code without performance degeneration due to quantization operations.

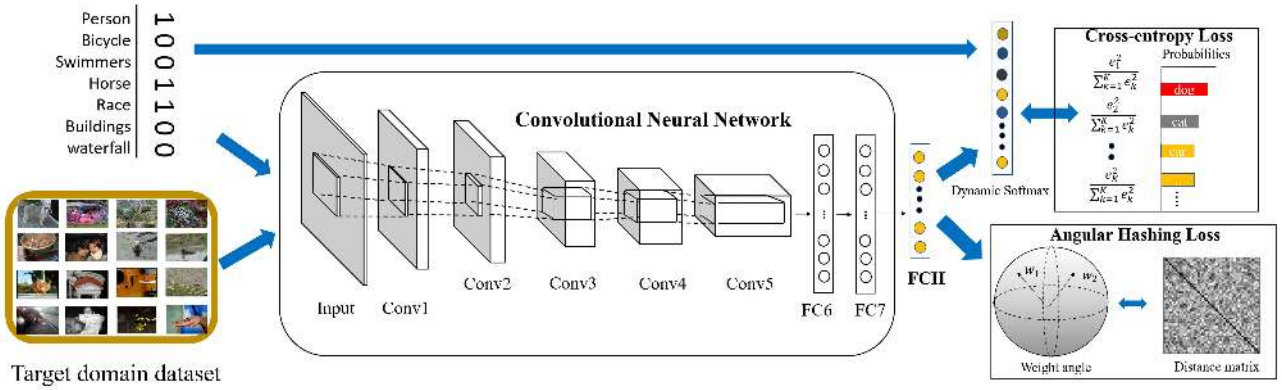


FIGURE 2. Overview of the end-to-end ADSH deep hashing framework, which comprised of four key components: (1) standard convolutional neural network (CNN), eg. AlexNet and ResNet, (2) the fully connected hash (FCH) layer for generating the k-dimensional feature representation, (3) dynamic Softmax for constraining learned features to be discriminative on a hypersphere manifold, and (4) the hashing distance matrix for adjusting the structure of deep feature in discrete space.

(3) A new training strategy is designed to solve the multi-label problem by creating a dynamic Softmax layer based on multi-label instances during training.

Experimental results show that this method can effectively improve image retrieval performance. More specifically, the proposed method not only inherits all the advantages of A-Softmax, but also considers the structure of learning features with angular margins between different classes. In addition, a clear geometric interpretation also contributes to the proposed loss. The proposed method can be considered as adjusting the feature structure of hyperspherical feature space.

The rest of this paper is organized as follows. In Section II, we use a toy example with geometric interpretation to present the proposed ADSH method and detail the proposed angular hashing loss and training method. The experimental results based on two well-known datasets are provided in Section III. Finally, conclusion is given in Section IV.

II. ANGULAR DEEP SUPERVISION HASHING (ADSH)

In this section, we first introduce the basic framework of the proposed ADSH method, and then use a toy example to visually analyze the deep feature distributions using different loss functions as well as proposing to adjust the angular direction to minimize incorrect binary code generation due to quantization. Inspired by the distribution observation, we developed an angular hashing loss function with consideration of Hamming distance matrix to improve the quality of the hash code. Finally, we designed a dynamic training strategy to support the training process for multi-label cases.

A. DEEP HASHING STRUCTURE

The main goal of the L2H methods is to improve the discriminative power of the hash feature. Intuitively, it is critical to minimize intra-class variations while keeping features of different classes separable. Unlike traditional supervised hashing methods that utilize pair-wise or triplet-wise labels as supervised information, the proposed ADSH focuses on point-wise based hashing methods that effectively use semantic information. The main idea of ADSH is to use CNN to achieve feature extraction, and the feature learning is constrained to be discriminative with angular margin. For mathematical representations, we denote the RGB image space as Ω, and the goal of ADSH is to learn CNN-based mapping $\mathcal{F}$ from Ω to k-bit binary code, $\mathcal{F} : \Omega \rightarrow \{+1, -1\}^k$.

The architecture of the proposed ADSH learning framework is shown in Figure 2. A conventional CNN such as AlexNet [29] or ResNet [30] with fully connected layers at the last few layers can be used as the backbone network. Moreover, the last fully connected layer is used as fully connected hash (FCH) layer for generating binary hash code by binarization process with use of *sign* function, which is defined as:

$$sign\,(x) = \begin{cases} 1, & if\ x \geq 0 \\ -1, & otherwise \end{cases} \tag{1}$$

On the other hand, the FCH layer is connected to a dynamic Softmax layer to output the probability of each class for calculating the cross-entropy loss. In ADSH, we propose to use the A-Softmax loss based on angular hash loss to achieve the discriminative power of features in continuous space. At the same time, we consider the relative position of the deep features in the Hamming distance space by adjusting the direction of each class. This approach not only enjoys effective training but also minimizes the effects of quantization errors with clear geometric interpretation. Moreover, the dynamic Softmax training strategy allows CNN to train directly on multi-label datasets using the gradient descent method. After training, the query images and the database images are encoded by network forward propagation, which makes the framework easy to implement in practical image retrieval systems.

B. SOFTMAX AND A-SOFTMAX FOR HASHING

In this subsection, we use a MINIST dataset [31] based toy example to demonstrate the advantages of A-Softmax loss compared to Softmax loss for hash code generation using the

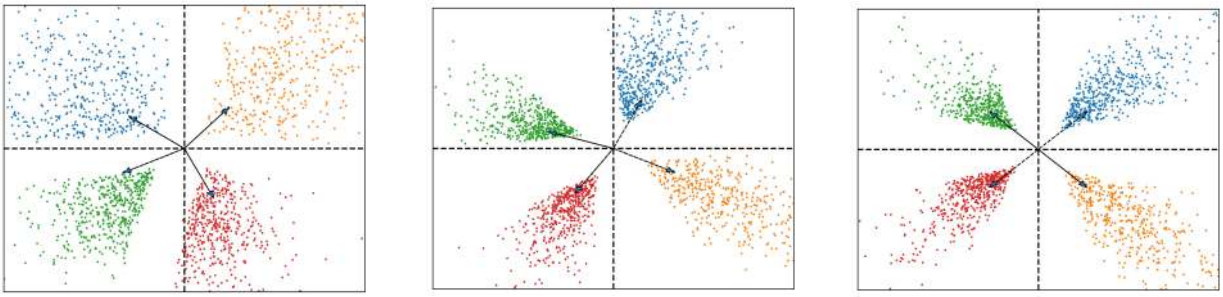


FIGURE 3. The distribution of deeply learned features under the supervision of (a) Softmax, (b) A-Softmax, and (c) A-Softmax with the fixed direction. The points with different colors denote features from different classes.

proposed ADSH learning framework. The importance of the feature distribution structure is also analyzed. Similar to the Center loss experiment of [32], we use LeNets++ network architecture with the last output layer reduced to 2 for feature visualization. Furthermore, we only selected four categories with total 4,000 images from the MNIST dataset instead of all 10 categories so as to demonstrate a simple example of 2-bit hash code generation with an upper limit of 4 classes. Therefore, we can visualize the quantization boundaries realized by the *sign* functions of these four classes on the 2D space. The Softmax loss is defined as

$$L_{softmax} = -\frac{1}{m}\sum_{i=1}^{m} log\frac{e^{w_{yi}^T x_i + b_{yi}}}{\sum_{j=1}^{C} e^{w_j^T x_i + b_j}} \tag{2}$$

where $x_i \in \mathbb{R}^k$ is an input deep feature with class label yi among the C classes. $w_{yi} \in \mathbb{R}^k$ is the yi-th column of Softmax layer weight $W \in \mathbb{R}^{k\times C}$, and $b_{yi} \in \mathbb{R}^1$ is the bias corresponding to class yi. Moreover, m and k denote the number of samples in min-batch and feature dimension, respectively. In order to compress intra-class features with strong constraints, many methods such as Center loss [32], A-Softmax [24] and L-Softmax [33] have been proposed. In this paper, we mainly study their application to hashing methods from the angular view. Therefore, we use A-Softmax to learn the features with large angular boundaries between classes. The A-Softmax loss is defined as

$$L_{A-softmax} = -\frac{1}{m}\sum_{i=1}^{m} \times log\frac{e^{\|x_i\|\psi(\theta_{yi,i})}}{e^{\|x_i\|\psi(\theta_{yi,i})} + \sum_{j=1,(j\neq yi)}^{c} e^{\|x_i\|\cos(\theta_{j,i})}} \tag{3}$$

where $\psi\left(\theta_{yi,i}\right) = (-1)^r \cos\left(\mu\theta_{yi,i}\right) - 2r$ and $\theta_{yi,i} \in \left[\frac{r\pi}{\mu}, \frac{(r+1)\pi}{\mu}\right]$ with $r \in [0, \mu - 1]$. Specifically, $\theta_{j,i}$ is the angle between vector w_j and x_i, and μ is the hyperparameter.

The deep feature distributions of MINIST dataset with only 4 classes generated by using Softmax and A-Softmax losses are plotted in Figure 3, which provides us with some interesting observations. Under the supervision of conventional Softmax loss, the learned features are well separated from each other as shown in Figure 3(a), but the intra-class variation is relatively large. In addition, some of the features of the red class crossed the quantization boundary (y-axis) and generated incorrect binary code (code for green class) after quantized by the *sign* function.

Figure 3(b) shows the feature distribution under the supervision of A-Softmax loss, which can achieve larger inter-class separation and compress inter-class variation. This distribution characteristic is better for hash code generation as compared with using Softmax loss. Unfortunately, some of the blue-class features crossed the *sign* function based quantization boundary (y-axis) with results of incorrect binary code generations. In addition, quite a lot of feature points of the blue, green and orange classes are distributed very close to the quantization boundaries. It is because the direct use of A-Softmax based supervision does not consider the difference between discrete space and continuous space. However, the intra-class features should keep away from the quantization boundaries to avoid or minimize incorrect binary code generation after the feature quantization process.

To address this issue, it is important to consider the relative angular direction of deep features, which can increase inter-class separability and keep the intra-class features further away from the quantization boundaries. Therefore, we propose to adjust the angular direction of A-Softmax loss to minimize incorrect binary code generation due to the quantization process. Figure 3(c) shows the feature distribution under the supervision of A-Softmax loss with angular direction correction, where the weights W of Softmax layer are set corresponding to the hash point in 2D space. It can be seen that the feature distribution is more separable, and each feature point can fall on a safe area with more buffered space from the quantization boundaries. Thus, this angular direction adjustment is possible to improve the quality of the hash code and the details of this A-Softmax loss with angular direction adjustment will be provided in the following subsection.

C. ANGULAR HASHING LOSS

1) HAMMING DISTANCE MATRIX

Basically, in this study, we want to develop an effective loss function to improve the discriminative ability of deep learning features and the intra-class feature performance in hash space. Let us consider the multi-class case first with a given training

FIGURE 4. **(a) An example of binary codes for three classes 'dog', 'cat', 'deer', (b) the distance matrices for each binary bit, and (c) the Hamming distance matrix *G* among those classes.**

set with C classes. We denote $b_i \in \{-1, 1\}^k$ as the representative k-bit hash code for the i-th class and b_i^j as the j-th bit in the hash code. Heuristically, we use Hamming distance matrix between classes as a metric to generate a discriminant hash code. In detail, the Hamming distance matrix G can be computed by XOR operations, where the entry of the matrix $G_{i,j}$ is the Hamming distance between b_i and b_j. This process is shown in Figure 4 with $C = 3$ as an example. Figure 4(a) lists the binary codes of these three classes (Dog, Cat and Deer). The distance matrix of each bit is illustrated in Figure 4(b), and the Hamming distance matrix G is shown in Figure 4(c). To maximize the distance between classes in the hash space, we maximize the value of each cell in the Hamming distance matrix, which represents the degree of difference between the two classes in the hash space. This is equivalent to maximizing the mean of the Hamming distance matrix or minimizing $mean\left(-G^{triu}\right)$, where G^{triu} is the upper triangular matrix of G. On the other hand, we also consider the balance of the matrix, because we find that the network can easily oppress some parts of the matrix, which may lead to redundancy in the hash code.

We discovered that in metric learning-based hashing methods, such as DSH [17], some of the hash bits are always 1 or -1 and do not contribute to the retrieval task. Therefore, we further minimize the variance of the matrix G to ensure that these binary codes can cover all matrix elements and there is no short board in the bucket theory. This criterion favors binary bit with an equal discriminability in the learning objective. These constraints are combined to generate better binary code and more discriminating bits for each class. Based on the two constraints, we defined a loss function L_M in discrete space as

$$\begin{aligned} L_M &= \alpha \cdot mean\left(-G^{triu}\right) + \beta \cdot variance\left(G^{triu}\right) \\ &= \alpha \cdot E_1 + \beta \cdot E_2 \end{aligned} \tag{4}$$

where E_1 encourages hash codes to be optimal in terms of distinctiveness, and E_2 ensures that the binary bits can be well balanced. The hyper-parameters α and β control the strength of regularizations in the final loss function. Note that Eq. (4) is the sum of losses that SGD can effectively minimize by backpropagation.

2) PARAMETER SHARING

In order to let the network learn the distribution for each representative hash code, we can simply pull the Euclidean distance between the weight of Softmax layer $\left\{w_{y_i}\right\}_{i=1}^{C}$ and another set of updateable parameters hash codes $\left\{b_{y_i}\right\}_{i=1}^{C}$. This is similar to Center loss [32] idea, but the proposed ADSH cast a view on angular margin. in which the network is jointly supervised through A-Softmax and hash matrix, these two loss functions share lots of common in learned deep feature space. In short, Softmax maximizes $W_{yi}^T x_i$, and our goal is to adjust the angular direction of W_{yi} by minimizing L_M. During the learning process, the weight W_{yi} will gradually approach the representative hash code b_{y_i} in term of angular distance. This implies that W_{yi} and b_{y_i} may have parallel directions in well-trained CNNs. In order to reduce parameter redundancy, we let the two losses share the same parameters, and b_{y_i} can be expressed as

$$b_{yi} = \tanh\left(\frac{W_{yi}}{\left\|W_{yi}\right\|}\right) \tag{5}$$

where *tanh* is a scaled *sigmoid* function to limit output range from -1 to 1, and $\frac{W_{yi}}{\left\|W_{yi}\right\|}$ is a unit-length vector pointing in the same direction as W_{yi}.

3) RELAXATION

The optimization in the Hamming distance matrix is not traceable due to the calculation of the matrix by XOR operations. Thus, we cannot update the parameters directly. This problem is solved by relaxing the constraints with the good relationship between Hamming distance and Euclidean distance. For example, given a pair of b_i and b_j with k dimension, we define the distance function as

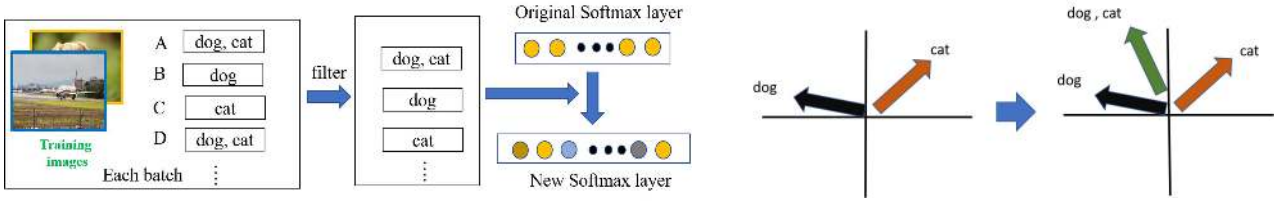
$$dist_H\left(b_i, b_j\right) = \frac{1}{2}\left(k - \left\langle b_i\ ,\ b_j\right\rangle\right) \tag{6}$$

where $dist_H\ (*, *)$ is Hamming distance between two binary codes, and $\langle *\ ,\ *\rangle$ is inner product between two vectors. With these modification, the network can be trained using a backpropagation algorithm with a mini-batch SGD method. The gradient of L_M with respect to final layer weight w_j can be computed as:

$$\frac{\partial L_M}{\partial w_j} = \alpha \cdot \frac{\partial E_1\left(W\right)}{\partial w_j} + \beta \cdot \frac{\partial E_2\left(W\right)}{\partial w_j} \tag{7}$$

$$\begin{aligned} \frac{\partial E_1\left(W\right)}{\partial w_j} &= \frac{\partial mean\left(-G^{triu}\right)}{\partial w_j} = \sum\nolimits_{i=1,(i\neq j)}^{C} \frac{\partial E_1\left(W\right)}{\partial G_{i,j}} \frac{\partial G_{i,j}}{\partial b_j} \frac{\partial b_j}{\partial w_j} \\ &= \sum\nolimits_{i=1,(i\neq j)}^{C} \frac{b_i}{C\left(C-1\right)} \left(1 - \tanh^2\left(\frac{W_j}{\left\|W_j\right\|}\right)\right) \frac{1}{\left\|W_j\right\|} \end{aligned} \tag{8}$$

$$\begin{aligned} \frac{\partial E_2\left(W\right)}{\partial w_j} &= \frac{\partial variance\left(G^{triu}\right)}{\partial w_j} = \sum\nolimits_{i=1,(i\neq j)}^{C} \frac{\partial E_2\left(W\right)}{\partial G_{i,j}} \frac{\partial G_{i,j}}{\partial b_j} \frac{\partial b_j}{\partial w_j} \\ &= \sum\nolimits_{i=1,(i\neq j)}^{C} \left(\frac{2G_{i,j}-1}{C(C-1)}\right) b_i \left(1 - \tanh^2\left(\frac{W_j}{\left\|W_j\right\|}\right)\right) \frac{1}{\left\|W_j\right\|} \end{aligned} \tag{9}$$



(a) (b)

FIGURE 5. The forward process of the proposed dynamic Softmax, (a) shows the generation of the new semantic weight, and (b) shows the multi-label case on 2-D dimension.

Then, we use joint supervision of A-Softmax and distance matrix for training. A-Softmax is used to train CNN for discriminative feature learning while Hamming distance matrix guides the relative position and direction of the deep features in the hash space. The overall loss function is defined as

$$\begin{aligned}\mathcal{L} &= L_{A-softmax} + L_M \\ &= L_{A-softmax} + \alpha \cdot mean\left(-G^{triu}\right) + \beta \cdot Variance\left(G^{triu}\right)\end{aligned} \tag{10}$$

where $\mathcal{L}_{A-softmax}$ and $\mathcal{L}_M$ are the A-softmax loss and the proposed loss in Eq.(4), respectively. This loss function is optimized by standard SGD as shown in Algorithm 1.

Basically, we summarize the learning details in CNN under joint supervision. Furthermore, the proposed method has a clear geometric interpretation as shown in Figure 3. A-Softmax provides good margins between classes, reducing coding errors in buffer space. In addition, Hamming distance matrix adjusts the position and angle of the features to maximize the discriminative power in the hash space.

Compared to the conventional metric learning-based hashing methods using pair-wise and triplet-wise training samples, the proposed ADSH method does not require a complex sample selection strategy. The angular based hashing approach learns the optimal hash code more directly because the learned features can be naturally converted to hash codes without quantization constraint in training process.

4) DYNAMIC A-SOFTMAX FOR MULTI-LABEL CASE

Basically, Softmax loss is designed for multi-class classification tasks with a single label as input. To perform Softmax loss on a multi-label, it may be necessary to put all the possibilities of inputting labels on the Softmax layer. But it is unrealistic due to the explosive growth of the number of class combinations. A common solution is to use sigmoid function to provide independent probabilities, but our goal is not to perform classification on the hashing space. We want to use A-Softmax to reduce the intra-class distance in the feature learning phase. Therefore, a dynamic Softmax layer is proposed to generate discriminative features for multi-label situation. Among them we dynamically change the number of neurons in the Softmax layer to fit the algorithm.

More specifically, given a multi-label dataset containing C different concepts, $X = \{x_n\}_{n=1}^m$ and $L = \{l_n\}_{n=1}^m$, in which m is the number of samples in a batch, are used to denote the deep feature and one-hot label vectors, respectively. There are two steps during batch optimization. We first filter the common labels by removing duplicated items from the label vector and count the number of non-repetitive labels as K. The Softmax layer can then be reconstructed to generate new semantic weights by looking up the mid-angle of each specified class. For example, an image x_n with the label "cat" and "dog" is given and then the combination of {cat, dog} is considered a new semantic center. The forward process of this proposed dynamic Softmax layer is illustrated in Figure 5, where the generation of the new semantic weights and the special multi-label case on 2D surface are shown in Figure 5(a) and 5(b), respectively. The new semantic weight $\widehat{w}_n$ is obtained as

$$\widehat{w}_n = \frac{1}{\sum_{i=1}^{C} l'_{ni}} \sum_{i=1}^{C} l'_{ni} w_{yi} \tag{11}$$

where $\{l'_n | n = 1, 2, \ldots, K\}$ is the new one-hot label vectors without duplicate item, $l'_{ni} = 1$ represents that i-th label is assigned to sample, w_{yi} is yi-th column of the original weight parameters $W = [w_1, w_2, \ldots, w_C]$, $\widehat{w}_n \in \mathbb{R}^d$ is n-th column of the new semantic weight parameters $\widehat{W} = [\widehat{w}, \widehat{w}_2, \ldots, \widehat{w}_K]$.

Algorithm 1 ADSH Training Algorithm

Input: Training data $\{x_i\}$. Initialized parameters θ in convolutional layers and parameters W in fully connected layers, weight in softmax layer w_{yi}
Output: The parameters θ, W, b_j, w_{yi}
Begin: The number of iterations $t \leftarrow 0$
While not converge **do**:
1. $t \leftarrow t1$
2. Calculate the representative binary code $b_{yi} = \tanh\left(\frac{w_{yi}}{\|w_{yi}\|}\right)$
3. Calculate the distance matrix $\left\{G_{i,j} | i = 1, 2, \ldots, C, j = 1, 2, \ldots, C\right\}$, where i, j represent the entry of i-th row and j-th column
4. Compute the joint loss by $\mathcal{L}^t = \mathcal{L}^t_{A-softmax} + \mathcal{L}^t_M$
5. Compute the backpropagation error
6. Update the parameters W
7. Update the parameters θ

End while

In addition, to make loss function to compute the corresponding probabilities based on new label instance, we need to modify the label of each sample. The new label $L = \{\widehat{l_n}\}_{n=1}^{m}$ is reassigned as follows:

$$\widehat{l_n} = z, s.t. l_z^{'} = l_n \tag{12}$$

With these settings, we can train the network like a normal classification task, except that the Softmax layer is generated based on the multi-label instance. In next section, we will show the proposed ADSH could achieve state-of-the-art performance on two well-known datasets for image retrieval application.

III. EXPERIMENTS

In this section, we will provide detailed information on the experiments to evaluate the image retrieval task of ADSH. In the first two experiments, we performed two evaluations using two datasets, CIFAR-10 and NUS-WIDE, with different sizes. The ablation experiment analyzed the impacts of hyperparameters and the performance of quantization control on the proposed loss function. In addition, extensive experiments were conducted to compare traditional methods with deep feature based methods. Finally, we visualize the high-dimensional distribution of deep features through t-SNE, which is a good explanation of the behavior of various hashing methods.

A. DATASET AND IMPLEMENTATION SETTING

We conducted experiments on two widely used benchmark datasets, CIFAR-10 [34] and NUS-WIDE [35]. CIFAR-10 is a multi-class dataset containing 60,000 32×32 color images with each image associated with only one label of 10 categories and each category containing 6,000 images. NUS-WIDE is a multi-label dataset with nearly 270,000 images from the web. Unlike CIFAR-10, each image in NUS-WIDE can be annotated with one or more labels with 81 semantic concepts. In our experiments, following the protocol in [17], we selected 198,512 images, of which 21 were the most commonly used concepts as our dataset. Moreover, each of these concepts contains at least 5,000 images. For the multi label dataset, the images sharing at least one label are considered as similar images.

The implementation uses pre-trained AlexNet [29] as the feature extractor and the FCH layer contains k nodes. For optimization, SGD with 0.001 initialized learning rate, 0.9 momentum and 0.0005 weight decay is used. Heuristically, set α and β are set to 1, while the hyperparameter μ of A-Softmax is set to 4. All experiments run on PyTorch platform.

B. BASELINE AND EVALUATION PROTOCOL

To demonstrate the effectiveness for image retrieval task, ADSH is compared with the well-known or state-of-the-art hashing methods. These methods can be roughly divided into two categories: (1) conventional hashing methods of SH [36], ITQ [10], FastH [37] and SDH [38], and (2) deep hashing methods of DNNH [20], DSH [17], DPSH [18], DLBHC [28], and HashNet [21]. For conventional hashing methods, 512-dimensional GIST descriptor from CIFAR-10 images and 1134-dimensional feature vector from NUS-WIDE images are used as input. The results from previous work [18] are used for comparison since the experimental settings are similar. For deep hashing methods, the results are obtained by running the source codes provided by their authors to train the model by ourselves. The pretrained AlexNet architecture is used as the backbone of the feature extraction, and resized raw image is directly feed into the network to generate the hash code.

Following the setting of [18], we conducted experiments under two settings with different number of training images. The code length k was set to 12 bits, 24 bits, 32 bits and 48 bits. Note that for the NUS-WIDE dataset, we return 5,000 samples to calculate MAP (MAP@5000) in the first experimental setup, and we use MAP @ 50000 in the second experimental setup.

C. RESULTS OF THE FIRST EXPERIMENT SETUP

In the first experimental setup, we randomly selected 100 images for each category in CIFAR-10 dataset. A total of 1,000 images are selected as query images. For the unsupervised hashing methods, all remaining images are used as training images. For supervised hashing methods, we randomly selected 5,000 images (500 per class) from the remaining images. The rest of the images are treated as database images. For NUS-WIDE dataset, 100 images per class, (i.e. 2,100 images), are randomly sampled as query images. Again, the rest images are used as training images for unsupervised hashing methods. For supervised hashing, we randomly selected a total of 10,500 images (500 per class) as training set for all baselines. The remaining images made up the dataset set.

Table 1 shows the overall image retrieval performance of the two datasets with different hash code lengths (12 bits, 24 bits, 32 bits, and 48 bits). We can observe that the deep hash methods are generally superior to the non-deep hash methods by a large margin. The significant performance improvements indicate that CNN-based image representation is advantageous than handcrafted feature representation. In terms of the deep hash method, ADSH method is always better than the competitor HashNet by about 4%. An interesting phenomenon is that the performance gap in the CIFAR-10 dataset is much larger than that in NUS-WIDE dataset. Specifically, ADSH can achieve a 12% performance improvement on CIFAR-10, compared to 0.4% performance improvement on NUS-WIDE, which are very impressive. This may be due to the small image size (32×32) of CIFAR-10, which is considered a challenging dataset. For multi-label dataset, we adopt the setting that image sample sharing at least one label are considered as similar image. It may result in smaller performance gap because it reduces the criteria for retrieval tasks. We also noticed the significant

TABLE 1. Mean average precision (MAP) results of different methods on both datasets under the first experiment setting, which return 5,000 top neighbors for NUS-WIDE.

Method	MAP of CIFAR-10				Method	MAP of NUS-WIDE			
	12 bits	24 bits	32 bits	48 bits		12 bits	24 bits	32 bits	48 bits
SH	0.127	0.128	0.126	0.129	SH	0.454	0.406	0.405	0.400
ITQ	0.162	0.169	0.172	0.175	ITQ	0.452	0.468	0.472	0.477
SDH	0.285	0.329	0.369	0.356	SDH	0.568	0.600	0.608	0.637
FastH	0.305	0.349	0.369	0.384	FastH	0.621	0.650	0.665	0.687
DBLHC	0.406	0.401	0.477	0.480	DBLHC				
DNNH	0.531	0.560	0.572	0.578	DNNH	0.656	0.701	0.720	0.683
DPSH	0.733	0.752	0.754	0.762	DPSH	0.761	0.794	0.801	0.812
DTSH	0.721	0.761	0.765	0.770	DTSH	0.773	0.802	0.813	0.827
DSH	0.714	0.733	0.731	0.739	DSH	0.737	0.748	0.751	0.740
HashNet	0.685	0.707	0.705	0.705	HashNet	0.770	0.802	0.806	0.816
ADSH	**0.754**	**0.780**	**0.786**	**0.795**	**ADSH**	**0.780**	**0.808**	**0.815**	**0.823**

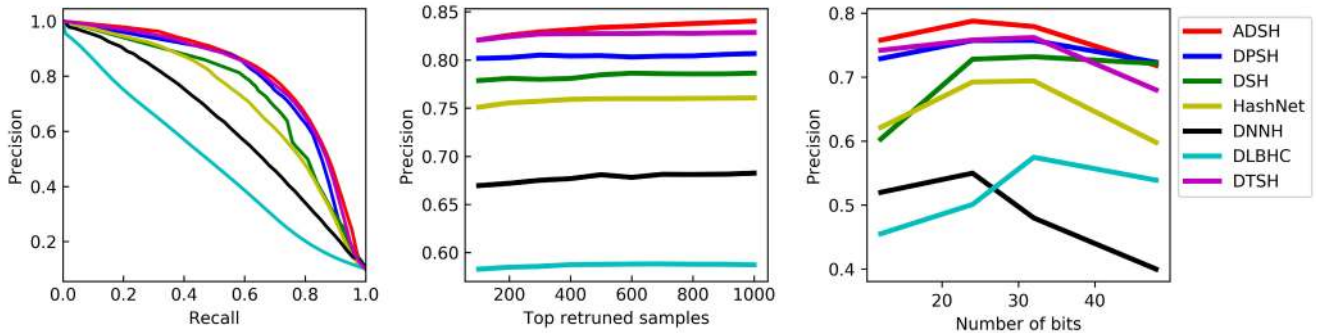


FIGURE 6. The comparison results of ADSH and other deep-hashing methods on the CIFAR-10 dataset under three evaluation matrices. (a) P-R curve at 48 bits (b) precision curve w.r.t top-N at 48bits (c) precision curve within hamming radius 2.

performance boost between ADSH and DLBHC, which also used Softmax loss to guide network training for larger image retrieval. Specifically, ADSH outperform DLBHC by 65% on CIFAR-10 dataset.

The performance of precision in terms recall, top return samples and number of bits are shown in Figure 6 and Figure 7 for CIFAR-10 and NUS-WIDE datasets, respectively. For recall of Figure 6(a) and Figure 7(a), it is clear that ADSH line is higher than other methods, which indicates that ADSH can achieve higher accuracy at the same recall level. This is ideal for practical systems that require high precision with a small amount of return samples. From Figure 6(b) and Figure 7(b), we can observe an increase in the precision of the top return samples.

The performance in terms of precision with different number of bits (code lengths) is shown in Figures 6(c) and 7(c). When the code length is 12 bits, ADSH achieves the highest P@H = 2 on all two datasets. This demonstrates that ADSH can learn more compact binary code for efficient image retrieval because each query requires only O(1) time for Hamming ranking at Hamming Radius 2. These results confirm the performance improvement of ADSH method over other well-known hashing methods.

D. RESULTS OF THE SECOND EXPERIMENT SETUP

The second experiment is designed to evaluate the performance of ADSH against the deep hashing methods under more training images. In CIFAR-10 dataset, 10,000 images (1,000 images per class) were selected as the query set, and the remaining 50,000 images were used as the training set and image database. In NUS-WIDE dataset, 2,100 images (100 images per class) were randomly sampled as test query images, and the remaining images (193,734 in total) were samples of training set and image database. The experimental results of the second experiment setup with the comparison between the deep hashing methods is shown in Table 2. It can be found that ADSH is still much better than all compared methods. In particular, ADSH method performed better than HashNet by approximately 2% on CIFAR-10 and 3% on NUS-WIDE. ADSH can also achieve better results than the DLBHC method because ADSH can maximize the semantic information in the hash space. Therefore, ADSH works well under different protocols.

E. EMPIRICAL ANALYSIS

1) IMPACTS OF THE HYPERPARAMETERS

In this subsection, we study the effects of hyperparameters α and β, where α controls the weight of the mean of Hamming

TABLE 2. Mean average precision (MAP) results of different methods on both datasets under second experiment setting, which return 50,000 top neighbors for NUS-WIDE.

Method	MAP of CIFAR-10				Method	MAP of NUS-WIDE			
	12bits	24bits	32bits	48bits		12bits	24bits	32bits	48bits
DLBHC	0.595	0.723	0.711	0.745	DLBHC				
DNNH	0.811	0.812	0.812	0.835	DNNH	0.766	0.771	0.772	0.776
DPSH	0.924	0.925	0.927	0.927	DPSH	0.810	0.823	0.825	0.831
DTSH	0.925	0.930	0.931	0.930	DTSH	0.801	0.825	0.831	0.833
DSH	0.922	0.923	0.922	0.919	DSH	0.770	0.775	0.776	0.779
HashNet	0.902	0.917	0.915	0.915	HashNet	0.770	0.796	0.805	0.811
ADSH	**0.929**	**0.933**	**0.940**	**0.936**	**ADSH**	**0.811**	**0.845**	**0.847**	**0.844**

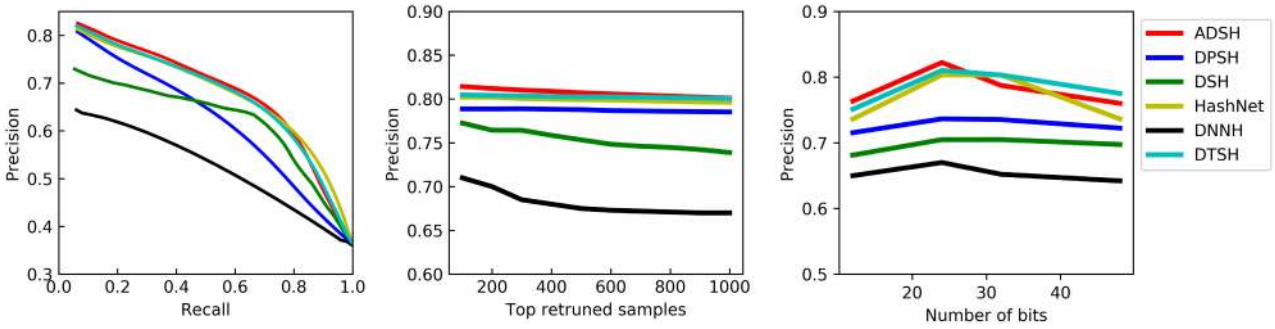


FIGURE 7. The comparison results of ADSH and other deep-hashing methods on the NUS-WIDE dataset under three evaluation metrics. (a) P-R curv at 48 bits (b) precision curve w.r.t top-N at 48bits (c) precision curve within hamming radius 2.

distance matrix, and β dominates the variance of Hamming matrix during the training process. All of this is important to the proposed ADSH model. Therefore, we conducted two experiments to illustrate the sensitivity of these two parameters to CIFAR-10. In the first experiment, we change the value of α from 0 to 100, where β is set to 0.1. Figure 8(a) shows the verification accuracy of these models. It is obvious that the accuracy has improved significantly from 0.692 with $\alpha = 0$, to 0.743 with $\alpha = 1$. However, when $\alpha = 100$, the performance drops to 0.721. The results show that Hamming distance matrix is helpful to achieve discriminative feature in the hash space.

In the second experiment, we changed the value of β from 0 to 100, and $\alpha = 1$. The results are shown in Figure 8(b). Different from the first experiment of this subsection, the performance of the model remains stable over a wide range of β and drops sharply to 0.69 at $\beta = 100$. As we have seen, the role of variance in Hamming distance matrix may be more like a generalization term used to balance the output of the network.

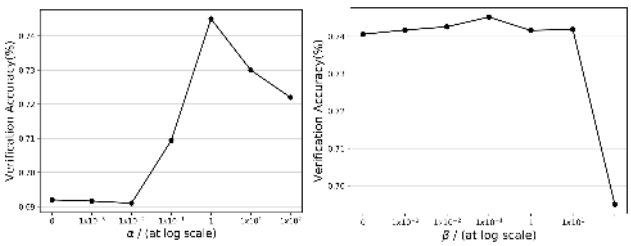


FIGURE 8. Accuracies on CIFAR-10, respectively achieve by (a) models adopt different α with fixed $\beta = 0.1$, (b) models with different value of β and fixed $\alpha = 1$.

2) COMPARISON WITH TRADITIONAL HASHING METHODS USING DEEP FEATURES

In order to validate the effectiveness of the proposed method and show that it does not only depends on the powerful feature extraction of CNNs, the proposed ADSH is compared with traditional methods by using deep features from pre-trained AlexNet model under the same experimental setup. We extract 4096-dimension feature from the penultimate layer as our input data. Table 3 compares the mean average precision (MAP) of various methods. It shows that ADSH can still achieve significant improvements on the CIFAR-10 dataset. The MAP of ADSH is about 25% higher than that of the best competitor FastH. It indicates that the advancement of our algorithm plays an important role in this image retrieval task.

3) EVALUATION OF QUANTIZATION ERROR

In this subsection, we gain a deeper understanding of the efficacy of quantization error control through joint supervision. We studied three semantic learning cases: (1) The

TABLE 3. Mean average Precision (MAP) result under the first experimental setting on CIFAR-10 dataset.

Method	MAP of CIFAR-10			
	12bits	24bits	32bits	48bits
SH+CNN	0.182	0.161	0.163	0.162
ITQ+CNN	0.272	0.288	0.290	0.296
SDH+CNN	0.492	0.566	0.593	0.602
FastH+CNN	0.578	0.635	0.656	0.667
ADSH	**0.754**	**0.780**	**0.786**	**0.795**

TABLE 4. Mean average precision (MAP) results of Softmax, A-Softmax, and ADSH on CIFAR-10 with 32 bits.

Methods	MAP without sign	MAP with sign
Softmax	0.514	0.482
A-Softmax	0.741	0.701
ADSH	0.767	0.786

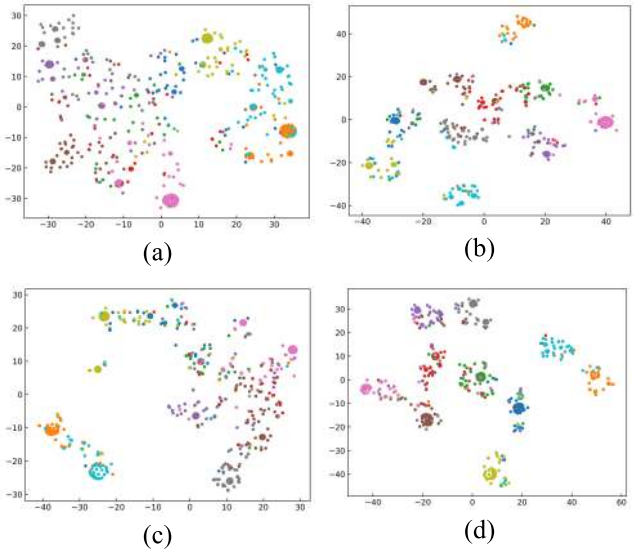


FIGURE 9. The t-SNE of hash codes learn from (a) Softmax, (b) A-Softmax, (c) HashNet, and (d) the proposed ADSH method.

network is only supervised by Softmax; (2) The network is only supervised by A-Softmax; (3) Supervise the network through joint supervision of the proposed method (ADSH). We perform these methods under the same network structure settings, and compute MAP with input of deep feature and hash code, respectively. For the deep features, we use cosine similarity as the distance metric and these results are shown in Table 4.

Through joint supervision under the A-Softmax and Hamming distance matrix, ADSH is much better than A-Softmax and Softmax in both cases. Interestingly, after feature binarization, the performance of MAP is reduced in Softmax and A-Softmax, but there is small performance boost in ADSH. Besides, we also notice that ADSH can achieve better result than A-Softmax in term of MAP without sign. This means that ADSH can even optimize the feature structure in the original feature domain. In general, the performance gap between deep features and hash codes verifies that ADSH method is an effective solution for controlling quantization errors.

4) VISUALIZATION OF HASH CODES

As mentioned earlier, if the training network only behaves under the supervision of Softmax like DLBHC, it will not be able to produce high quality hash codes. To understand this situation more clearly, we compare Softmax loss, A-Softmax loss, HashNet and the proposed ADSH methods with t-SNE visualization of hash codes. Specifically, we train these four methods on the same CIFAR-10 dataset. Figure 9 shows the visualization of the four methods. We can observe that the hash code generated by Softmax loss is more similar to the random distribution of data points. For A-Softmax, although the hash codes from different categories are well separated, it does not have the clear structure as ADSH. In addition, ADSH generates more compact hash codes than HashNet. This verifies that the hash code generated by the proposed ADSH method is more discriminative, enabling image retrieval to be performed more efficiently.

IV. CONCLUSION

In this paper, we proposed a new image retrieval hashing method based on A-Softmax loss, called ADSH (Angular Deep Supervision Hashing). In order to reduce the gap between continuous space and discrete space, we used angular hashing loss to optimize the deep features in the hash space with joint supervision of Hamming distance matrix and A-Softmax loss. This new loss function not only enjoys effective training, but also minimizes the effects of quantization errors, with clear intuition and geometric interpretation as shown in a toy example. In addition, we also proposed a dynamic Softmax training strategy to address training problem on multi-label datasets. Compared to the conventional metric learning-based hashing method using pair-wise and triplet-wise training samples, ADSH method does not require a complex sample selection strategy. Comprehensive experiments confirmed that ADSH encourages the network to generate more compact hash codes, resulting in the most advanced image retrieval performance on the CIFAR-10 and NUS-WIDE benchmark datasets.

In the future, we plan to enhance the dynamic Softmax concept in order to improve the results for multi-label datasets. In addition, the benefits of metric learning and semantic learning can be combined into a single end-to-end training network.

REFERENCES

[1] J. Eakins and M. Graham, "Content-based image retrieval," JISC Techol. Appl. Programme, Univ. Northumbria, Newcastle, Newcastle upon Tyne, U.K., Tech. Rep. 39, Oct. 1999.
[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[3] X. Wang, T. X. Han, and S. Yan, "An HOG-LBP human detector with partial occlusion handling," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 32–39.

[4] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 117–128, Jan. 2011.

[5] J. Wang, O. Kumar, and S. Chang, "Semi-supervised hashing for scalable image retrieval," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3424–3431. doi: 10.1109/CVPR.2010.5539994.

[6] 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.*, Jun. 2012, pp. 2074–2081.

[7] B. Kulis and T. Darrell, "Learning to hash with binary reconstructive embeddings," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1042–1050.

[8] J. Wang, H. T. Shen, J. Song, and J. Ji, "Hashing for similarity search: A survey," 2014, *arXiv:1408.2927*. [Online]. Available: https://arxiv.org/abs/1408.2927

[9] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proc. VLDB*, 1999, vol. 99, no. 6, pp. 518–529.

[10] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2916–2929, Dec. 2013.

[11] K. He, F. Wen, and J. Sun, "K-means hashing: An affinity-preserving quantization method for learning binary compact codes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2938–2945.

[12] M. Norouzi and D. M. Blei, "Minimal loss hashing for compact binary codes," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 353–360.

[13] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, "Supervised hashing for image retrieval via image representation learning," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 1–7.

[14] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 815–823.

[15] W. Chen, X. Chen, J. Zhang, and K. Huang, "Beyond triplet loss: A deep quadruplet network for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 403–412.

[16] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. BMVC*, 2015, vol. 1, no. 3, p. 6.

[17] 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.

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

[19] X. Wang, Y. Shi, and K. M. Kitani, "Deep supervised hashing with triplet labels," in *Proc. Asian Conf. Comput. Vis.* New York, NY, USA: Springer, 2016, pp. 70–84.

[20] 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.

[21] Z. Cao, M. Long, J. Wang, and P. S. Yu, "HashNet: Deep learning to hash by continuation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2017, pp. 5608–5617.

[22] Y. Cao, M. Long, J. Wang, H. Zhu, and Q. Wen, "Deep quantization network for efficient image retrieval," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 1–7.

[23] H. Zhu, M. Long, J. Wang, and Y. Cao, "Deep hashing network for efficient similarity retrieval," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 1–7.

[24] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 212–220.

[25] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "NormFace: L_2 hypersphere embedding for face verification," in *Proc. 25th ACM Int. Conf. Multimedia*, 2017, pp. 1041–1049.

[26] X. Zhe, S. Chen, and H. Yan, "Deep class-wise hashing: Semantics-preserving hashing via class-wise loss," 2018, *arXiv:1803.04137*. [Online]. Available: https://arxiv.org/abs/1803.04137

[27] H.-F. Yang, K. Lin, and C.-S. Chen, "Supervised learning of semantics-preserving hash via deep convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 437–451, Feb. 2018.

[28] K. Lin, H.-F. Yang, J.-H. Hsiao, and C.-S. Chen, "Deep learning of binary hash codes for fast image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2015, pp. 27–35.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," presented at the 25th Int. Conf. Neural Inf. Process. Syst., Lake Tahoe, NV, USA, vol. 1, 2012.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[31] Y. LeCun. *The MNIST Database of Handwritten Digits*. Accessed: Nov. 1998. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[32] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer, 2016, pp. 499–515.

[33] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *Proc. ICML*, 2016, vol. 2, no. 3, p. 7.

[34] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009, vol. 1, no. 4.

[35] 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.

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

[37] 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.

[38] Q. Li, Z. Sun, R. He, and T. Tan, "Deep supervised discrete hashing," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2482–2491.

CHANG ZHOU received the B.Sc. degree from Donghua University, Shanghai, China, in 2016, and the master's degree from the City University of Hong Kong, Hong Kong, in 2017, where he is currently pursuing the Ph.D. degree with the Department of Electronic Engineering. His current research interests include computer vision and deep learning.

LAI-MAN PO (M'92–SM'09) received the B.S. and Ph.D. degrees in electronic engineering from the City University of Hong Kong, Hong Kong, in 1988 and 1991, respectively.

He has been with the Department of Electronic Engineering, City University of Hong Kong, since 1991, where he is currently an Associate Professor with the Department of Electrical Engineering. He has authored over 150 technical journal and conference articles. His current research interests include image and video coding with an emphasis on deep learning-based computer vision algorithms.

Dr. Po is a member of the Technical Committee on Multimedia Systems and Applications and the IEEE Circuits and Systems Society. He was the Chairman of the IEEE Signal Processing Hong Kong Chapter, in 2012 and 2013. He was an Associate Editor of *HKIE Transactions*, in 2011 to 2013. He also served on the Organizing Committee of the IEEE International Conference on Acoustics, Speech, and Signal Processing, in 2003, and the IEEE International Conference on Image Processing, in 2010.

WILSON Y. F. YUEN received the B.Sc. degree (Hons.) in applied computing from the University of Hertfordshire, U.K., and the M.Sc. degree in information engineering from The Chinese University of Hong Kong. He has established himself as an expert in information engineering and user experience design, and a pioneer in multimedia and game development education, over 20 years. His passion toward innovative technology is well-recognized and was named one of Debrett's Hong Kong 100 Most Influential People (one out of ten individuals under technology and digital category).

He was the Chief Information Officer at Commercial Radio Hong Kong, and taught post-graduate courses at The Hong Kong Polytechnic University, City University of Hong Kong, and Hong Kong Baptist University. In 2010, he founded TFI Digital Media Ltd., a HK-based technology company specializing in video-related technologies. Meanwhile, he also provides advisory services to multiple family offices, which are listed on Forbes Hong Kong's Top 50, with interests across real estate, telecom, and hospitality industries.

He currently serves as a Professor of practice with the Academy of Film, Hong Kong Baptist University, a City University Department of Electronic Engineering Advisory Committee Member, a HP Enterprise OEM Customer Advisory Board Member, a Mentor and Vetting Committee of PolyU Micro Fund, and the committee of Qualifications Framework Industry Training Advisory Committee (ITAC) under Hong Kong Education Bureau. He is also an Honorary Advisor of the Professional Information Security Association and a Global Design Ambassador of the Interaction Design Foundation.

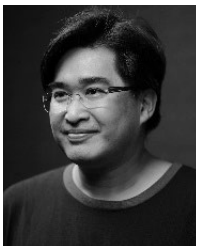
KWOK WAI CHEUNG (M'10) received the B.Eng., M.Sc., and Ph.D. degrees from the City University of Hong Kong, in 1990, 1994, and 2001, respectively, all in electronic engineering. He was with Hong Kong Telecom, as an Engineer, from 1990 to 1995. He was a Research Assistant with the Department of Electronic Engineering, City University of Hong Kong, from 1996 to 2002. He was an Assistant Professor with the Chu Hai College of Higher Education, Hong Kong, from 2002 to 2016. He has been with the School of Communication, Hang Seng University of Hong Kong, as an Associate Professor, since 2016. His current research interests include image processing, machine learning, and social computing.

XUYUAN XU received the B.E. and Ph.D. degrees in electronic engineering from the City University of Hong Kong, in 2010 and 2014, respectively. He is currently with Tencent Video, Tencent Holdings Limited, as a Senior Engineer. His current research interests include 3D video coding, 3D view synthesis, audio/video fingerprint, and object detection.

KIN WAI LAU received the B.E. degree (Hons.) in information engineering from the City University of Hong Kong, in 2017, where he is currently pursuing the Ph.D. degree with the Department of Electronic Engineering. He is currently with TFI Digital Media Ltd., as a Software Engineer. His current research interests include image and video processing, video/image retrieval, computer vision, and deep learning.

YUZHI ZHAO received the B.Eng. degree in electronic information from the Huazhong University of Science and Technology, Wuhan, China, in 2018. He is currently pursuing the Ph. D. degree with the Department of Electronic Engineering, City University of Hong Kong. His current research interests include image processing, computer vision, deep learning, and machine learning.

MENGYANG LIU received the B.E. degree in optoelectronic engineering from the Shanghai University of Electric Power, Shanghai, China, in 2014, and the M.Sc. degree with a dissertation in electronic and information engineering from the City University of Hong Kong, Hong Kong, in 2015, where he is currently pursuing the Ph.D. degree with the Department of Electronic Engineering. His current research interests include image and video processing, video indexing, computer vision, and machine learning.



PETER H. W. WONG (SM'08) received the B.Eng. degree (Hons.) in computer engineering from the City University of Hong Kong, in 1996, and the M.Phil. and Ph.D. degrees in electrical and electronic engineering from The Hong Kong University of Science and Technology (HKUST), in 1998 and 2003, respectively. He was a Postdoctoral Fellow with the Department of Information Engineering, The Chinese University of Hong Kong, from 2003 to 2005. He was with the Applied Science and Technology Research Institute Company Ltd., as a Member of the Professional Staff, from 2005 to 2007. From 2007 to 2008, he was a Visiting Assistant Professor with the Department of Electronic and Computer Engineering, HKUST. From 2008 to 2015, he was with Visual Perception Dynamics Labs (Mobile) Ltd., involved in RGBW display technology. He is currently the Chief Software Engineer of TFI Digital Media Ltd., involved in high dynamic range and wide color gamut video processing. He is the author of about 40 technical journal and conference articles. He was the inventor of ten US patents.