IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 20, 2018, accepted January 7, 2019, date of publication January 10, 2019, date of current version February 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2891894

Deep Attention-Guided Hashing

ZHAN YANG[1], OSOLO IAN RAYMOND[1], WUQING SUN[1], AND JUN LONG[1,2]
[1]School of Information Science and Engineering, Central South University, Changsha 410083, China
[2]Network Resources Management and Trust Evaluation Key Laboratory of Hunan Province, Central South University, Changsha 410083, China

Corresponding author: Jun Long (junlong@csu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61472450, in part by the Key Technology Research and Development Program of Hunan Province under Grant 2018GK2052, and in part by the Science and Technology Plan of Hunan under Grant 2016TP1003.

ABSTRACT With the rapid growth of multimedia data (e.g., image, audio, and video) on the Web, the learning-based hashing techniques, such as deep supervised hashing, have proven to be very efficient for large-scale multimedia search. The recent successes seen in the learning-based hashing methods are largely due to the success of the deep learning-based hashing methods. However, there are some limitations to the previous learning-based hashing methods (e.g., the learned hash codes containing repetitive and highly correlated information). In this paper, we propose a novel learning-based hashing method, named deep attention-guided hashing (DAgH). DAgH is implemented using two stream frameworks. The core idea is to use the guided hash codes which are generated by the hashing network of the first stream framework (called the first hashing network) to guide the training of the hashing network of the second stream framework (called the second hashing network). Specifically, in the first network, it leverages an attention network and hashing network to generate the attention-guided hash codes from the original images. The loss function we propose contains two components: the semantic loss and the attention loss. The attention loss is used to punish the attention network to obtain the salient region from pairs of images; in the second network, these attention-guided hash codes are used to guide the training of the second hashing network (i.e., these codes are treated as supervised labels to train the second network). By doing this, DAgH can make full use of the most critical information contained in images to guide the second hashing network in order to learn efficient hash codes in a true end-to-end fashion. Results from our experiments demonstrate that DAgH can generate high-quality hash codes and it outperforms the current state-of-the-art methods on three benchmark datasets: CIFAR-10, NUS-WIDE, and ImageNet.

INDEX TERMS Supervised learning-based hashing, attention-guided strategy.

I. INTRODUCTION

In recent years, the amount of multimedia data (text, image, audio and video data) has been growing exponentially. In order to solve the problems of huge storage requirements and learning capacity in dealing with big data, hashing has been the most popular technique for effective binary representation in many tasks due to its fast retrieval and storage efficiency. Generally speaking, the hashing technique, a widely-studied solution to approximate nearest neighbor search, aims to map the original high-dimensional features to a low-dimensional representation, or a short code, called hash code. Then, re-ranking these short codes (hash codes) in response to each query task, requires only a few computations of the Hamming distance operation for efficient multimedia retrieval (i.e., the hashing technique can use a few Bytes to encode one image of several MBytes or one video of several GBytes). Due to the advantages above, hashing has been applied to many large-scale image retrieval [1]–[4], text-image cross-model retrieval [5], and person re-identification tasks [6]. There are two categories of hashing: data-independent and data-dependent hashing. In this paper, we will build data-dependent hashing methods for generating high quality hash codes, which can capture the potential image representations to achieve better performance than data-independent hashing methods, e.g., Spectral Hashing (SH) [7].

Data dependent methods can be further categorized into supervised and unsupervised methods. Unsupervised methods retrieve the neighbors under some kinds of distance metrics, e.g., Iterative Quantization (ITQ) [8]. Compared to the unsupervised methods, supervised methods utilize the semantic labels to improve performance. Many researchers

have demonstrated that labels of datasets can improve the quality of hash codes and achieve some success along this direction, e.g., Supervised Hashing with Kernels (KSH) [9], Distortion Minimization Hashing (DMS) [4], Minimal Loss Hashing (MLS) [11], Order Preserving Hashing (OPH) [12], Hamming Distance Metric Learning [13], Semantic Hashing [14], Supervised Discrete Hashing (SDH) [15]. However, the quality of hash codes generated is highly dependent on the way feature selection is done, and these methods use hand-crafted features for representation. The need to perform manual feature selection has been a big limitation to the success of these methods.

In the last few years, deep learning networks (e.g., convolutional neural networks) have been shown to have powerful feature extraction capabilities in image processing. They are able to extract high-level features, which leads to attaining much higher performance levels than using hand-crafted features in many image tasks. To solve the limitations of traditional data-dependent hashing methods, this paper focuses on a learning-based hashing method that adopts deep neural networks as the nonlinear functions to enable end-to-end learning of learnable representations and hash codes. These learning-based methods [10], [16]–[18], which use pairwise labels to jointly learn similarity-preserving representations and optimize the pair-wise loss and quantization loss, have exhibited high performance on many benchmark tests.

Although recent learning-based hashing methods have achieved significant progress in multimedia retrieval, there are some limitations of previous learning-based hashing methods in generating long hash codes (say, more than 24 bits), e.g., the learned long hash codes contain repetitive and highly correlated information. Any natural image will contain some useless information, or some interference information that is not relevant for a particular task. Images of the same category may contain completely different backgrounds, different categories of images may have similar backgrounds, directly generating the hash codes by a standard learning-based method (as shown in Figure 1) will in practice result in a higher possibility of having correlated bits as the length of the hash codes increases. Then, highly correlated bits have a large impact on retrieval performance (i.e., the cost-performance ratio decreases with increase in the length of the hash code). As an extreme example, if 256-bit hash codes are positively and negatively completely correlated, the performance will be similar to that of the 1-bit hash codes. To solve these limitations, in this paper, we deal with the salient regions and backgrounds of the images separately. Specifically, the main idea of the paper is to firstly adopt an attention network to generate the attention images from the original images, (i.e, use visual attention models to localize regions in an image to capture features of the regions) and then use pairwise information to generate the attention-guided hash codes from the attention images. Secondly, we use these attention-guided hash codes to guide the training of the second hashing network (i.e., these codes are treated as supervised labels to train the second network).

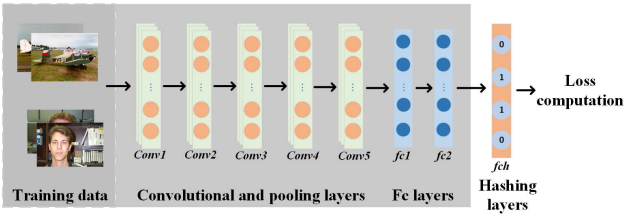


FIGURE 1. The basic architecture of supervised learning-based hashing.

The contributions of this work are summarized as follows:

1) The proposed **DAgH** model combines two stream frameworks. The first stream framework consists of an attention network and a hashing network (the first hashing network). The role of the first stream framework is to generate the attention-guided hash codes. A novel method of using the semantic loss and attention loss to train the first stream framework is proposed. The second stream framework contains another hashing network i.e. the second hashing network. This hashing network is guided by the attention-guided hash codes which were generated from the first stream framework. The second stream framework is trained by the proposed guide loss. To the best of our knowledge, this is the first learning-based hashing method that uses its own attention-guided hash codes to guide the training of the original image hashing network.
2) In order to guarantee the quality of the final hash codes and eliminate the quantization error, the **DAgH** model uses a continuous activation function to ensure that the first stream framework is a true end-to-end network and a threshold activation function to ensure that the second stream framework directly generates the final hash codes. In the first stream framework, we chose to use a continuous ATanh activation function for training because it's easier to optimize than using a *sign* function with no extra quantization loss. As a result, it shows stronger capacity in learning high quality attention-guided hash codes. When the second stream framework is trained by the attention-guided hash codes, we can use a *sign* activation function to constrain the output of the second stream framework for generating the binary codes directly. These operations trade off efficacy for efficiency.

The remainder of this paper is structured as follows. In Section II, we briefly introduce the related works. In Section III, we highlight the motivation of our method and provide some theoretical analysis for its implementation. In Section IV, we introduce our experimental results and corresponding analysis and finally in Section V conclude the paper.

II. RELATED WORK

A. HASHING

By representing multimedia data as binary codes and taking advantage of fast query retrieval, hashing is a novel technique

that can resolve the information retrieval problems in this multimedia era. Wang *et al.* [21] have provided a comprehensive literature survey that covers the most important methods and latest advances in image retrieval.

We can divide the hashing methods into two categories: data-independent and data-dependent methods. In the early researches, Locality Sensitive Hashing (LSH) was one of the data-independent methods used. LSH hashes input items so that items that are similar have a high probability of being mapped to the same ''buckets'' (the number of buckets being much smaller than the universe of possible input items) [22]. LSH and several variants (e.g., kernel LSH [23] and p-norm LSH [24]) are widely used for large-scale image retrieval. However, there are many limitations of data-independent methods, e.g., the efficiency is low and it requires longer hash codes to attain high performance. Due to the limitations of the data independent methods, current researchers focus on using a variety of machine learning techniques to learn more efficient hash functions based on a given dataset.

Data dependent methods can be further categorized into supervised, semi-supervised and unsupervised methods. Unsupervised hashing methods learn hash functions that encode data points to binary codes by training from unlabeled data. Typical learning criteria include minimize reconstruction error [25]–[28] and graph structure learning [29], [30]. Iterative Quantization (ITQ) is one of the unsupervised methods in which the projection matrix is optimized by iterative projection and thresholds according to the given datasets [8]. Compared to the semi-supervised and unsupervised methods, supervised methods utilize the semantic labels to improve performance. Many researchers have proposed along this direction and have achieved some success (They have demonstrated that labels of datasets can improve the quality of hash codes), e.g., Supervised Hashing with Kernels (KSH) [9], Distortion Minimization Hashing (DMS) [4], Minimal Loss Hashing (MLS) [11], Order Preserving Hashing (OPH) [12], Hamming Distance Metric Learning [13], Semantic Hashing [14], Supervised Discrete Hashing (SDH) [15]. The hash codes are generated by minimizing the Hamming distance between similar pairs and maximizing the Hamming distance between dissimilar pairs.

Recently, deep convolutional neural networks have yielded amazing results on many computer vision tasks, this success has attracted the attention of researches of learning-based hashing methods. Convolutional Neural Network Hashing (CNNH) is one of the early works to use a learning-based hashing method, which utilize two stages to learn the image features and hash codes. Following this work, many learning-based hashing techniques have been proposed, e.g., Deep Semantic Ranking Hashing (DSRH) [31] which learns the hash functions by preserving semantic similarity between multi-label images. Deep Visual-Semantic Quantization (DVSQ) [1] generates the compact hash codes by optimizing an adaptive margin loss and a visual-semantic quantization loss over multi-networks. Deep Supervised Hashing (DSH) [32] designs a loss function to pull the outputs of similar pairs of images together and pushes the dissimilar ones far away. Its outputs are relaxed to real values to avoid optimizing the non-differentiable loss function in Hamming distance. Network In Network Hashing (NINH) [33] adopts a triplet ranking loss to capture the relative similarities of images. Deep Supervised Discrete Hashing (DSDH) [34] uses both pairwise label and classification information to learn the hash codes under a single steam framework. Guo *et al.* [35] show that existing DSH can achieve good results with short hash codes (e.g., 8 to 24 bits) but only lead to marginal performance gain with long hash codes (e.g., 128 bits). They try to divide a single network into many sub-networks to generate hash codes respectively. Extensive researches have taken advantage of deep learning techniques to achieve great improvements compared to traditional data-dependent hashing methods.

However, existing learning-based methods do not consider the high correlation problem of long hash codes. Although convolutional neural networks have powerful capabilities in image feature extraction, they do not deal with the irrelevant features in the image. When long hash codes need to be generated, the correlation problem of the hash codes cannot be ignored. In this paper, we introduce a high-quality hash code generation method, where an attention network is embedded to mine salient regions for guiding the standard supervised learning-based hashing framework.

B. SALIENT REGIONS LEARNING

The key challenge of learning high quality hash codes is to locate the salient regions in images. Many methods for locating salient regions have been proposed in recent years. Previous methods of locating the salient regions can be categorized into traditional methods and deep learning based methods.

Traditional methods include techniques such as [36]–[38] locating the salient regions by unsupervised methods. Following these works, some hashing methods locate salient regions to improve performance in the unsupervised manner. Shen *et al.* [39] proposed a cross-modal hashing method which uses RPN [40] to detect salient regions. Then, the two cross-modal networks are used to encode the region information, the semantic dependencies and cues between the words. DPH [41] uses GBVS [42] to count the scores for each pixel. Then, a collection of salient regions are generated based on increasing threshold scores. However, traditional methods use ready-made models to locate salient regions and therefore, when encountering a new dataset, there is no guarantee that the learned salient regions are accurate.

Due to the success of deep learning, most of the methods depend on powerful deep features, which have shown a higher performance gain than hand-crafted features on image classification [43]–[45], [47]. Zhao *et al.* [48] adopted similarity labels to train part model for person re-identification. Lin *et al.* [49] proposed a bilinear structure, which computes the pairwise feature interactions by two-stream convolutional neural networks to capture the different salient regions

between input images and achieved high performance in bird classification. RA-CNN [50] is a recurrent-attention convolutional neural network, which can discover salient regions and learn region-based features recursively. Motivated by Fu *et al.* [50] and Jin [51], we adopt a novel attention network to generate the attention image and use a hashing network to learn the attention-guided hash codes. Then use the generated attention-guided hash codes to guide the second hashing network to learn the final hash codes.

III. DEEP ATTENTION-GUIDED HASHING

In this section, we first give the problem formulation, then show the details of our proposed method, including the framework, loss function and training strategy, and finally show its extensions to out-of-sample data.

A. PROBLEM FORMULATION

In similarity retrieval systems, we are given a training set $\mathcal{X} = \{x_i\}_{i=1}^{N}$, each image represented by a d-dimensional feature vector $x_i \in \mathbb{R}^d$, where $\mathcal{X} \in \mathbb{R}^{d \times N}$. In supervised learning-based hashing, the pairwise information $S = \{s_{ij}\}$[1] is derived as:

$$s_{ij} = \begin{cases} 1, & \text{if images } x_i \text{ and } x_j \text{ share same class label} \\ 0. & \text{otherwise} \end{cases} \quad (1)$$

Supervised learning-based hashing method learn a nonlinear function $f : \mapsto b \in \{-1, 1\}^K$ from an input space $\mathbb{R}^d$ to Hamming space $\{-1, 1\}^K$ with deep neural network.

[1]Note that one image may belong to multiple categories.

This method generally contains three steps: 1) using a network for learning deep features of each image x_i, 2) using a fully-connected hashing layer (*fch*) for transforming the deep features into K-dimensional continuous representation $\omega_i \in \mathbb{R}^K$, 3) using a *sign* function to quantize the continuous representation ω_i into K-bit binary hash code $b_i \in \{-1, +1\}^K$. The similarity labels $S = \{s_{ij}\}$ can be constructed from semantic labels of data points or relevance feedback in real retrieval systems. In addition, the threshold function $sign(\cdot)$ is an element-wise *sign* function defined as follows:

$$sign(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ -1. & \text{otherwise} \end{cases} \quad (2)$$

B. NETWORK ARCHITECTURE

To address the limitations of previous learning-based hashing methods, we propose a novel learning-based method. Figure 2 shows the proposed **DAgH** model. Our method includes two stream framework. The first stream contains an attention network and a hashing network. The attention map is the most critical part of our network, since it allows the network to know which regions should be focused on. Then the hashing network uses the attention images to generate the attention-guided hash codes. Thereafter, in the second stream, these attention-guided hash codes are treated as supervised labels to train an image hashing network which can generate the final hash codes for the input images. The details of all stream frameworks are described in the following subsections.

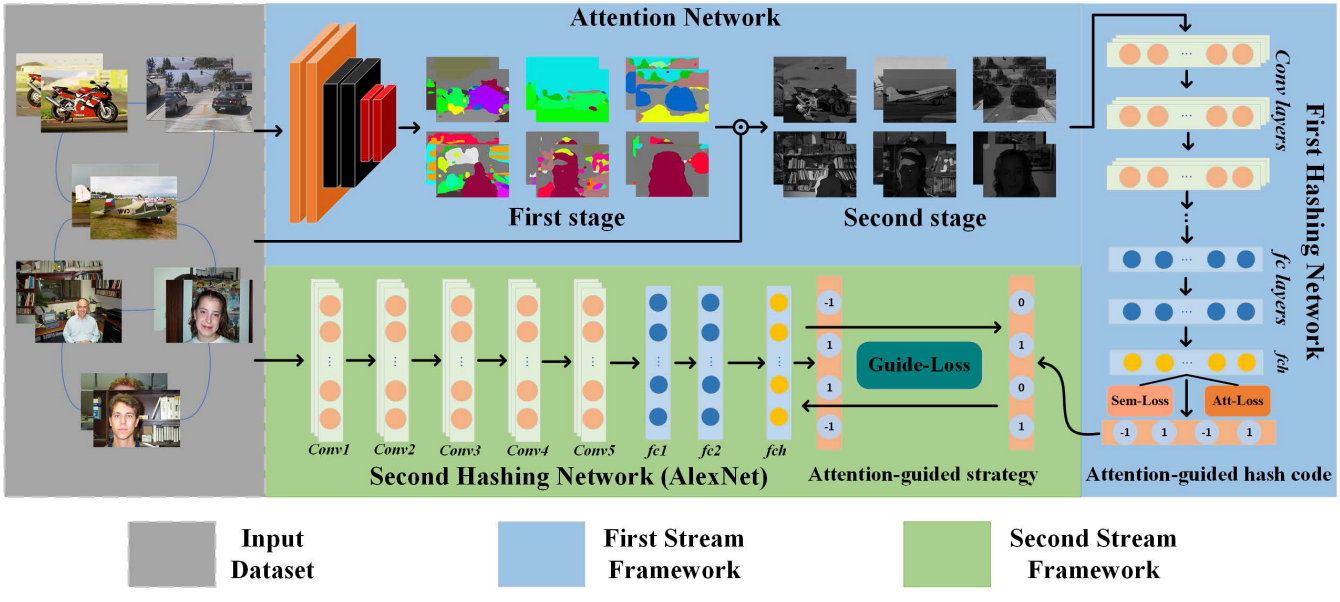


FIGURE 2. The proposed architecture for deep attention-guided hashing (DAgH). DAgH consists of two stream frameworks: 1) the first stream framework contains an attention network based on FCN-16 network for learning the attention image pair, the attention processing contains two stages. Then, the hashing network uses AlexNet (or ResNet) for learning the attention-guided hash codes. The first stream framework consists of two loss functions: the semantic loss and the attention loss, and uses a continuous ATanh activation function for training. 2) the second stream framework contains a hashing network that adopts AlexNet for learning hash codes, it then uses the attention-guided hash codes for its supervised labels, and the final hash codes are generated directly by the second stream framework using *sign* activation function. (Best viewed in color.)

C. THE FIRST STREAM FRAMEWORK

As mentioned previously, this stream framework is a true end-to-end deep model which includes an attention network, i.e., $Attention(\cdot|\Theta_a)$, and a hashing network, i.e., $Hash(\cdot|\Theta_1)$. In this subsection, we will introduce the main processes of the two networks and the details of the network training.

1) THE ATTENTION NETWORK

The role of the attention network is to find salient regions in the original image that need to get attention. These regions are the most representative of the theme of the image, so they can be used for better salient region restoration, and for the first hashing network to focus the assessment on. This attention processing consists of two stages. In the first stage, the proposed FCN-based attention network [62] is used to map the original input image pair $[x_i, x_j]$ to the preliminary attention image pair $[\hat{x}_i, \hat{x}_j]$. Inspired by Jin [51], to build a learnable attention network and guarantee that it generates accurate attention images, we define a normalization function to restrict the value of each pixel between 0 and 1:

$$norm_i(p, q) = \frac{x_i(p, q) - \min(x_i)}{\max(x_i) - \min(x_i)}, \tag{3}$$

where (p, q) denotes the location (p, q) in an image x, $\max(x)$ denotes the maximum pixel value in an image x, $\min(x)$ denotes the minimum pixel value in an image x.

In the second stage, the attention image pair $[\tilde{x}_i, \tilde{x}_j]$ is computed through a Hadamard product $\otimes$ of the original image pair $[x_i, x_j]$ and the normalization function of the preliminary attention image pair $[\hat{x}_i, \hat{x}_j]$:

$$[\tilde{x}_i, \tilde{x}_j] = [x_i, x_j] \otimes norm_{[\hat{x}_i, \hat{x}_j]}(p, q). \tag{4}$$

Then we can encode the attention image pair $[\tilde{x}_i, \tilde{x}_j]$ by the first hashing network. The attention network can be gradually fine-tuned their parameters through (10) to mine salient regions automatically.

2) THE FIRST HASHING NETWORK

The generated attention image pair $[\tilde{x}_i, \tilde{x}_j]$ as the input of the first hashing network and the semantic information as the pairwise label to train the first hashing network. After the first hashing network is trained, the attention-guided hash code $B^{att} = \{b_i^{att}\}_{i=1}^N$ is calculated through the trained hashing network:

$$b_i^{att} = sign(Hash(\tilde{x}_i|\Theta_1)), \tag{5}$$

where b_i^{att} is the attention-guided hash code, Θ_1 denotes the parameters of the first hashing network,[2] and $\tilde{x}_i$=$Attention$ $(x_i|\Theta_a)$ is the attention image that is input into the first hashing network and generated by the attention network, Θ_a is the parameters of the attention network.

[2]Note that, the ATanh activation function(the detail about the ATanh activation function can be found in (11)) is used to train the first stream framework (8) and the *sign* activation function is used as the final output of the first stream framework.

The no-quantization loss training strategy of the first stream framework is expatiated as follows:

Similarity Measure

For a pair of binary hash codes b_i and b_j, the relationship between their Hamming distance dist_H and inner product $\langle \cdot, \cdot \rangle$ is formulated as follows: $\text{dist}_H = \frac{1}{2}(K - \langle b_i, b_j \rangle)$. The larger the inner product of two hash codes, the smaller the Hamming distance, and vice versa. Therefore, the inner product through two hash codes is a reliable criterion for evaluating the similarity between them.

In supervised learning-based hashing method, the Maximum Likelihood (WL) estimation of the hash codes $B = [b_1, b_2, \ldots, b_N]$ for all N images is:

$$\log P(S|B) = \prod_{s_{ij} \in \mathcal{S}} \log P(s_{ij}|B), \tag{6}$$

where $P(S|B)$ denotes the likelihood function. Given each image pair with their similarity label $([x_i, x_j], s_{ij})$, $P(s_{ij}|b_i, b_j)$ is the conditional probability of s_{ij} given the pair of corresponding hash codes $[b_i, b_j]$, which is naturally defined as logistic function:

$$P(s_{ij}|b_i, b_j) = \begin{cases} \sigma(\langle b_i, b_j \rangle), & s_{ij} = 1 \\ 1 - \sigma(\langle b_i, b_j \rangle), & s_{ij} = 0 \end{cases} \tag{7}$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid activation function, $\langle b_i, b_j \rangle = \frac{1}{2} b_i^T b_j$.

Loss Function

Semantic Loss Considering the similarity measure,the following loss function is used to learn the hash codes:

$$\begin{aligned} \mathcal{L}_{sem} &= -\log P(S|B) = -\sum_{s_{ij} \in S} \log(s_{ij}|B) \\ &= -\sum_{s_{ij} \in S} (s_{ij} \langle b_i, b_j \rangle - \log(1 + \exp(\langle b_i, b_j \rangle))), \end{aligned} \tag{8}$$

where $b_i = sign(\omega_i)$, which converts the K-dimensional representation ω_i to exactly binary hash codes.[3] Equation (8) is the negative *log* likelihood loss function, which represents the Hamming distance of two similar images that are as small as possible, and the Hamming distance of two dissimilar images that are as large as possible. Then, we define an attention loss to train the attention network to capture some salient regions of the image.

3) ATTENTION LOSS

In training the attention network, we denote the continuous representation pair of *fch* layer (also called binary-like codes) as $[\omega_i, \omega_j]$. Then we obtain the optimal hash code pair $[b_i, b_j]$ from the continuous representation pair $[\omega_i, \omega_j]$. Given $[b_i, b_j] \in \{-1, 1\}^k$, the cosine similarity between the continuous representation pair can be defined as $\cos(\omega_i, \omega_j) = \frac{\omega_i^T \omega_j}{||\omega_i||_2 ||\omega_j||_2}$, which is in the range of $(-1, 1)$.

[3]Note that, Equation (8) need to first learn the continuous representation ω_i, which are quantized to binary values in a separated operation using *sign* function, this will result in quantization errors.

Therefore, we use $\frac{\cos(\omega_i,\omega_j)+1}{2}$ to restrict the similarity value to (0, 1). The attention loss is written as below:

$$\mathcal{L}_{att} = \sum_{i,j} ||S_{ij} - \frac{\cos(\omega_i, \omega_j) + 1}{2}||_2 + \sum_{i,j} \max(0, \lambda - ||S_{ij} - \frac{\cos(\omega_i, \omega_j) + 1}{2}||_2), \quad (9)$$

where $\lambda > 0$ is a margin parameter. The attention loss will punish the attention network to make it better capture the salient regions.

Overall, combining Equation (8) and (9), the loss of the first stream framework can be written as:

$$\min_{\Theta_a, \Theta_1} \mathcal{L}_{sem} + \nu \mathcal{L}_{att}, \quad (10)$$

where Θ_a, Θ_1 are the first stream framework parameters efficiently optimized using standard back-propagation with automatic differentiation techniques.

End-to-End Learning

In many of the recent hashing methods [8], [15], [16], [32], [34], [51]–[54], quantization error is an important part of their optimization process, which will directly result in retrieval quality. These hashing methods first need to learn continuous representations (binary-like codes) through *sigmoid* and tanh functions, then, the binary-like codes are binarized into hash codes in a separate operation of *sign* thresholding. Therefore, the gap between the binary-like codes and hash codes is called the quantization error. For examples, in [51], the quantization error is defined as $\mathcal{L}_{reg} = \sum_i ||\omega_i - b_i||_1$, in ITQ [8], the quantization error is defined as $\mathcal{L}_{ITQ} = ||\omega_i - b_i||_2$, where $b_i = sign(\omega_i) \in \{1, -1\}^K$. Although the optimization methods propose to reduce the quantization error, the activations of the *fch* layer are still not binary. This is because a *sign* function is non-smooth and non-convex, and therefore has no gradient (i.e., the gradient of *sign* function is zero for all non-zero inputs, which makes the classical back-propagation infeasible for training deep networks.). Cao *et al.* [55] proposed a justifiable approach based on the continuation of the tanh function, which approaches the *sign* function with the scale parameter β in its limit: $\lim_{\beta\to\infty} \tanh(\beta x) = sign(x)$, they prove the convergence of this optimization when adopting a sequence of increasing values of β during training. However, in order to ensure that the continuous tanh function is differential everywhere that can be optimized via standard back-propagation, a regularization term should be considered [56]. Such activation function is named as Adaptive Tanh (ATanh):

$$b_i = \tanh(\beta\omega_i) + \epsilon\, ||\frac{1}{\beta}||_2^2, \quad (11)$$

where ϵ is the regularization constant. The second term of (11) is a regularization term. The regularization term is a penalty to the standard $\tanh(\beta x_i)$, when β gradually increases, the ATanh function approaches the *sign* function and has the reliable-ability to generate hash codes. When $\beta \to \infty$, the optimization problem will converge to the original deep learning to hash problem in (8) with $sign(x)$ activation function. We follow the empirical parameters setting and first set the parameter $\beta_0 = 1$ as the initialization. At each epoch T, we increase β and fine-tune the first stream framework of **DAgH** in the next epoch. With the parameter $\beta \to \infty$ of the (11), the network will converge to the first stream framework of **DAgH** with *sign* as activation function, which can generate high-quality attention-guided hash codes as we required. The time consumption of ATanh as the activation function in the whole network is negligible (i.e., both forward and backward computation is negligible) [56]. Different from the previous hashing methods mentioned above, there is no-extra quantization error within such an end-to-end hashing net, hence it shows stronger capacity in learning high-quality attention-guided hash codes.

Algorithm 1 Deep Attention-Guided Hashing (**DAgH**)

Input Training Image pair with their similarity label $([x_i, x_j], s_{ij})$ in the first stream framework, a sequence $1 = \beta_0 < \beta_1 < \beta_2 \ldots < \beta_m = \infty$. Training Image x_i and the attention-guided hash codes B^{att} in the second stream framework. Training epoches T_1 and T_2 of the first and second stream framework optimizations, respectively.

Output First stream framework: $sign(Hash(Attention(x_i|\Theta_a)|\Theta_1))$; Second stream framework: $sign(Hash(x_i|\Theta_2))$.

Begin Construct the pairwise information matrix S according to (1).

1. **for** $t = 1 : T_1$ epoch **do**
2. Compute $[b_i^{att}, b_j^{att}]$ according to (5)
3. Train the first hashing network (8) with (11) as activation
4. Compute Θ_a, Θ_1 according to (10)
5. Set converged the first stream framework as next epoch initialization
6. **end for**
7. **return** $sign(Hash(Attention(x_i|\Theta_a)|\Theta_1))$, $\beta_m \to \infty$.

1. **for** $t = 1 : T_2$ epoch **do**
2. Compute $\tilde{y}_i^2$ according to $\tilde{y}_i^2 = Hash(x_i|\Theta_2)$
3. Compute Θ_2 according to (14)
4. Set converged the second stream framework as next epoch initialization
5. **end for**
6. **return** $sign(Hash(x_i|\Theta_2))$.

D. THE SECOND STREAM FRAMEWORK

As shown in Figure 2, we directly adopt a pre-trained AlexNet as the base of the second hashing network. After obtaining the attention-guided hash code B^{att}, we thereafter utilize it as the supervised labels and the original images $\mathcal{X} = \{x_i\}_{i=1}^N$ to train the hashing network. When the hashing network is trained, the final hash codes b_i^f are computed through the trained hashing network:

$$b_i^f = sign(Hash(x_i|\Theta_2)), \quad (12)$$

where b_i^f is the final hash code, Θ_2 denotes the parameters of the second hashing network, and $\tanh(\cdot)$ as activation of the *fch* layer of the hashing network.

The details of learning strategy are explained as follows:

Attention-Guided Strategy

As mentioned above, when the first stream framework is trained, the generated attention-guided hash codes are used as the supervised labels to guide the second hashing network. Considering the powerful image feature extraction ability of convolutional neural networks, here we adopt the famous and widely used AlexNet, which is commonly used in baseline models. The AlexNet consists of 5 convolutional layers ($c1 - c5$), and 2 fully-connected layers ($fc1 - fc2$), and is pre-trained on the ImageNet dataset. To obtain the hash codes, we add a k-nodes hash layer, called *fch*, each node of *fch* layer corresponds to 1 bit in the target hash code. With the *fch* layer, the previous layer representation is transformed to a k-dimensional representation. The architecture of the second hashing network is shown in Figure 2.

More specifically, let $\tilde{y}_i^2 = Hash(x_i|\Theta_2)$ be the output of the second hashing network, where x_i is the original input image and Θ_2 is the parameter of the second hashing network. Since our goal is to use the attention-guided hash code b_i^{att} to guide the second hashing network through sigmoid cross-entropy loss function, we need to convert the value of -1 in the attention-guided hash codes to 0 so that the value of the attention-guided hash code is $b_i^{att} \in \{0, 1\}^K$. We define the following likelihood functions:

$$P(b_{ik}^{att}|\tilde{y}_{ik}^2) = \begin{cases} \sigma(\tilde{y}_{ik}^2), & b_{ik}^{att} = 1 \\ 1 - \sigma(\tilde{y}_{ik}^2), & b_{ik}^{att} = 0 \end{cases} \tag{13}$$

where b_{ik}^{att} is the hash code corresponding to the k-th bit of the i-th element in b_i^{att}, $\tilde{y}_{ik}^2$ is the output of the k-th node in *fch* layer of the i-th element, and $\sigma(\cdot)$ is a sigmoid function as shown in (7).

Loss Function

Guide Loss In order to use the attention-guided hash codes to guide the second hashing network, we define a guide loss, which is written as follows:

$$\begin{aligned} \mathcal{L}_g &= -\frac{1}{KN}\sum_{k=1}^{K}\sum_{i=1}^{N}\log P\left(b_{ik}^{att}|\tilde{y}_{ik}^2\right) \\ &= -\frac{1}{KN}\sum_{k=1}^{K}\sum_{i=1}^{N}[\log {P_{ik}}^{b_{ik}^{att}} \cdot \log(1-P_{ik})^{(1-b_{ik}^{att})}] \\ &= -\frac{1}{KN}\sum_{k=1}^{K}\sum_{i=1}^{N}\left[b_{ik}^{att}\log P_{ik} + \left(1-b_{ik}^{att}\right)\log\left(1-P_{ik}\right)\right], \end{aligned} \tag{14}$$

where N is the number of training images, K is the number of bits in each hash code, and $P_{ik} = \sigma(\tilde{y}_{ik}^2)$.

In order to minimize (14), we use the Back-Propagation (BP) algorithm to learn the parameter Θ_2 of the second hashing network with stochastic gradient descent (SGD). Specifically, we take the derivative of the guide loss:

$$\begin{aligned} \frac{\partial \mathcal{L}_g}{\partial \tilde{y}_{ik}^2} &= \frac{\partial \mathcal{L}_g}{\partial P_{ik}}\frac{\partial P_{ik}}{\partial \tilde{y}_{ik}^2} \\ &= -\frac{1}{KN}(b_{ik}^{att}\frac{1}{P_{ik}} - \frac{1-b_{ik}^{att}}{1-P_{ik}})(P_{ik}(1-P_{ik})) \\ &= -\frac{1}{KN}(P_{ik} - b_{ik}^{att}). \end{aligned} \tag{15}$$

Thereafter, we can obtain $\partial\mathcal{L}_g/\partial\Theta_2$ with $\partial\mathcal{L}_g/\partial\tilde{y}_{ik}^2$ using the chain rule, i.e., we can use BP to update the parameter Θ_2 of the second hashing network. After training, we obtain the trained AlexNet model for the final hashing model and the corresponding image hash codes can be generated by (12).

E. OUT-OF-SAMPLE EXTENSION

After our proposed **DAgH** model is trained, we can easily generate its hash code through the second hashing network. For example, given a new instance $x_q \notin \mathcal{X}$, we directly use it as the input of **DAgH** model, then forward propagate the second hashing network to generate its hash code as follows:

$$b_q = sign(Hash(x_q|\Theta_2)). \tag{16}$$

IV. EXPERIMENTS

In order to demonstrate the performance of our proposed **DAgH** method, we carried out extensive experiments on three widely used benchmark datasets, i.e., CIFAR-10, NUS-WIDE, and ImageNet, to verify the effectiveness of our method.

A. DATASETS AND SETTINGS

CIFAR-10 [57] dataset consists of 60,000 images with a resolution of 32×32 in 10 categories (each category has 6,000 images). Each image has only one category. In our experiment, we randomly selected 100 images per category (i.e., 1,000 images in total) as the test set, 500 images per category (i.e., 5,000 images in total) as the training set. The rest of the images are used as gallery in the testing phase.

NUS-WIDE [58] is a dataset contains that nearly 270K (260,648) images collected from the public web. It is a *multi-label* dataset. There are 81 semantic concepts manually annotated for evaluating retrieval performance. In our experiment, as in [17] and [34], we selected the 21 most frequent concepts. We randomly sample 100 images per class (i.e., 2,100 images in total) as the test set, 500 images per class (i.e., 10,500 images in total) as the training set. The rest of the images are treated as the gallery in the testing phase.

ImageNet [59] dataset is a well-known benchmark dataset for the Large Scale Visual Recognition Challenge (ILSVRC 2015). It contains 1,000 categories with over 1.2M images in the training set and 50,000 images in the validation set, where each image has only one category. As in [3] and [54], we randomly selected 100 categories which led to a database with about 120K images and a query set with about 5,000 images.

In this dataset, we randomly selected 100 images per class (i.e., 10,000 in total) as the training set.

B. BASELINES

We compared our proposed **DAgH** method against some classic or state-of-the-art hashing methods. We roughly divided these methods into two groups: traditional hashing methods and learning-based hashing methods. The traditional hashing methods include unsupervised hashing methods: **SH** [7], **ITQ** [8], and supervised hashing methods: **SDH** [15], **KSH** [9]. The learning-based hashing methods include **DPSH** [52], **DHN** [16], **CNNH** [17], **DNNH** [33], **DSDH** [34]. These methods are based on either AlexNet [19] or CNN-F [60] network architecture. The AlexNet network and CNN-F network have similar network architectures (i.e., They consist of 5 convolutional layers and 2 fully connected layers). As in the traditional hashing methods, we used DeCAF$_7$ features [61]. For learning-based methods, we used raw images as input. In fact, in the past few years, many more advanced networks have been created such as ResNet [20], WRNs [46]. The aim of our paper is to demonstrate a novel technique based on AlexNet that is able to outperform baseline models. If we adopted the advanced networks, we would be unable to know whether the performance gain was given by our **DAgH** method or by the advanced networks.

We evaluated the image retrieval quality on four metrics: mean Average Precision (**mAP**), Precision-Recall curves (**PR**), Precision curves within Hamming distance 2 (**P@H=2**), Precision curves with different Number of top returned samples (**P@N**). For fair comparison, we adopted MAP@1000 for ImageNet and MAP@5000 for other datasets as in [34]

C. IMPLEMENTATION DETAILS

The **DAgH** method was implemented on Pytorch and batch gradient descent was used to train the network. As shown in Figure 2, our model consists of three networks: an attention network and two hashing networks. We use a very famous attention network, i.e., FCN [62] as the base model for the attention network. As discussed in [62], there are three different network models (i.e., FCN-8s, FCN-16s, and FCN-32s). We use the fusing method of FCN-16s to improve performance. Readers can find more details about the attention network in [62]. We used AlexNet for the all hashing networks. We fine-tuned convolutional layers and fully-connected layers copied from AlexNet pre-trained on ImageNet and trained the hashing layer *fch* by back-propagation (BP). As the *fch* layer is trained from scratch, we set its learning rate to be 10 times that of the lower layers. In our proposed **DAgH** method, in batch form are used as the input and every two images in the same batch constitute an image pair. The parameters of our proposed **DAgH** model are learned by minimizing the proposed loss function. The training procedure, i.e., **DAgH**, is summarized in Algorithm 1.

Network Parameters: In our **DAgH**, the value of hyper-parameter ν is 50 and λ is 0.3. The parameter ϵ of ATanh follows the empirical value of 0.001 in [56]. We use mini-batch Stochastic Gradient Descent (SGD) with 0.9 momentum and the learning rate annealing strategy implemented in Pytorch. The mini-batch size chosen was 32 and the weight decay parameter selected was 0.0005.

D. RESULTS AND DISCUSSIONS

The mAP results of all methods for different lengths of hash codes on CIFAR-10, NUS-WIDE, and ImageNet are listed in Table 1. Results on CIFAR-10 dataset show that the proposed **DAgH** method substantially outperforms all other methods against which it was compared. Compared to traditional hashing methods, such as, ITQ, the best shallow hashing method using deep features achieves an absolute boost of 77.83%, 78.25%, 78.74%, and 78.68% corresponding to different lengths of hash codes, respectively. In addition, most of the learning-based hashing methods perform better than the traditional hashing methods. In particular, DSDH, the state-of-the-art learning-based hashing method, achieves the best performance among all the learning-based methods. Compared to DSDH, our **DAgH** method can achieve absolute

TABLE 1. mean Average Precision (mAP) of hamming ranking for different number of bits on the three image datasets.

Method	CIFAR-10				NUS-WIDE				ImageNet			
	12 bits	24 bits	32 bits	48 bits	12 bits	24 bits	32 bits	48 bits	12 bits	24 bits	32 bits	48 bits
SH [7]	0.127	0.128	0.126	0.129	0.454	0.406	0.405	0.400	0.185	0.273	0.328	0.395
ITQ [8]	0.162	0.169	0.172	0.175	0.452	0.468	0.472	0.477	0.305	0.363	0.462	0.517
SDH [15]	0.285	0.329	0.341	0.356	0.568	0.600	0.608	0.637	0.253	0.371	0.455	0.525
KSH [9]	0.303	0.337	0.346	0.356	0.556	0.572	0.581	0.588	0.136	0.233	0.298	0.342
DHN [16]	0.555	0.594	0.603	0.621	0.708	0.735	0.748	0.758	0.269	0.363	0.461	0.530
CNNH [17]	0.429	0.511	0.509	0.522	0.611	0.618	0.625	0.608	0.237	0.364	0.450	0.525
DNNH [33]	0.552	0.566	0.558	0.581	0.674	0.697	0.713	0.715	0.219	0.372	0.461	0.530
DPSH [52]	0.713	0.727	0.744	0.757	0.752	0.790	0.794	0.812	0.143	0.268	0.304	0.407
DSDH [34]	0.726	0.762	0.785	0.803	0.743	0.782	0.799	0.816	0.312	0.353	0.481	0.533
DAgH	**0.731**	**0.777**	**0.809**	**0.821**	**0.753**	**0.791**	**0.811**	**0.825**	**0.322**	**0.377**	**0.503**	**0.551**

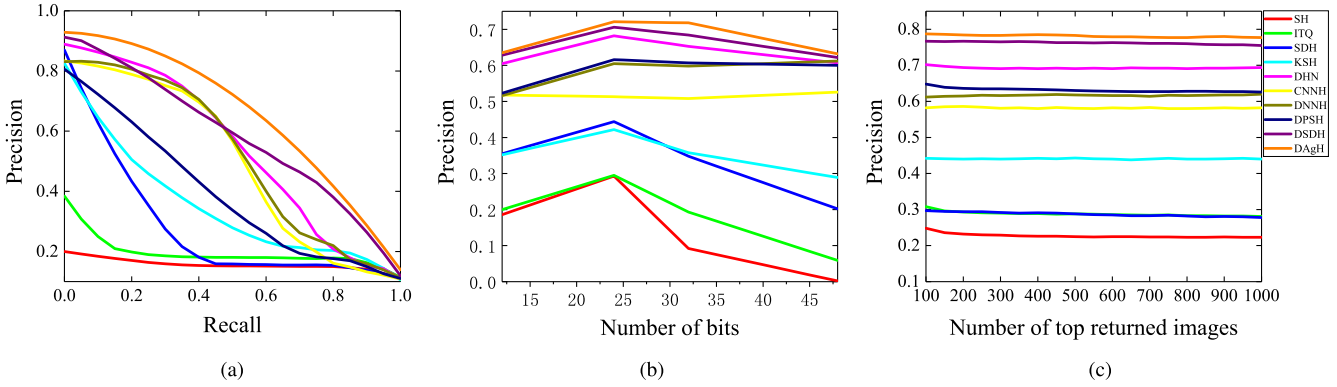


FIGURE 3. The results of DAgH and comparison methods on the CIFAR-10 dataset under three evaluation metrics. (a) Precision-Recall curve @ 48 bits. (b) Precision within Hamming radius 2. (c) Precision curve w.r.t. top-*n* @ 48 bits.

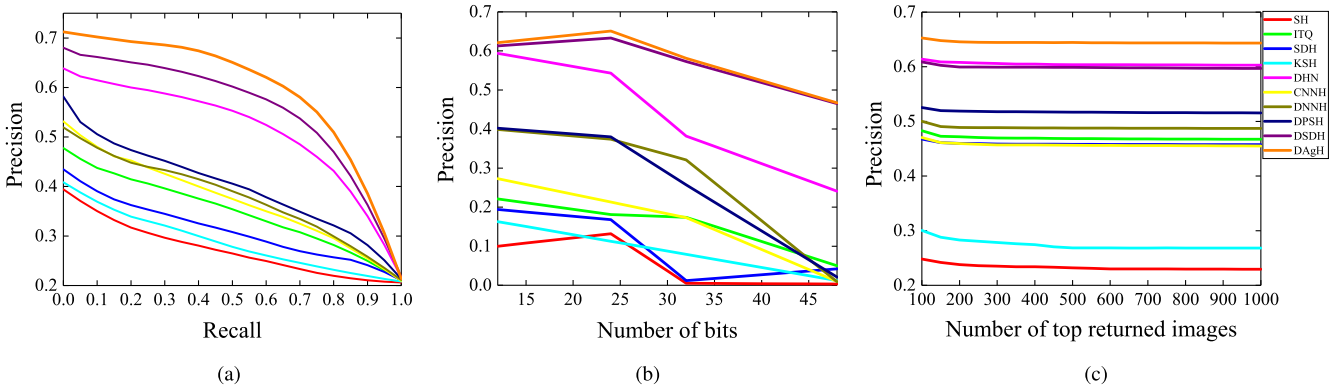


FIGURE 4. The results of DAgH and comparison methods on the NUS-WIDE dataset under three evaluation metrics. (a) Precision-Recall curve @ 48 bits. (b) Precision within Hamming radius 2. (c) Precision curve w.r.t. top-*n* @ 48 bits.

boosts of 0.68%, 1.90%, 2.9%, and 2.2% in average mAP corresponding to different lengths of hash codes, respectively. Similar to the other hashing methods, we also conducted experiments for large-scale image retrieval. For NUS-WIDE and ImageNet datasets, if two images share at least one same label, they are considered to belong to the same category. The results of experiments using the NUS-WIDE and ImageNet datasets on Table 1 show that the proposed **DAgH** method outperforms the best existing traditional hashing image retrieval methods (i.e., ITQ) by 41.20% and 5.17% in average mAP for different lengths of hash codes on these datasets, respectively. Compared to the state-of-the-art learning-based hashing method (i.e., DSDH). We achieve absolute boosts of 1.26% ,3.64% in average mAP for different lengths of hash codes on these datasets, respectively. These results demonstrate that our approach can boost the retrieval performance.

We also observe from the Table 1 that the gap between the learning-based methods and traditional hashing methods is larger on CIFAR-10 dataset than NUS-WIDE and ImageNet datasets. The reasons are that the number of categories in NUS-WIDE and ImageNet datasets are more than those in CIFAR-10 dataset, and each of the image may contain multiple labels. By carefully comparing the performance of different bits, we found that our proposed method showed a higher degree of performance improvement when tested on at long bits (i.e., 32bits and 48 bits) compared to short bits (i.e, 12bits and 24 bits). This means that our approach can make the hash codes more informative.

An important indicator for evaluating image retrieval performance is Precision within Hamming radius 2 (**P@H=2**) because such Hamming ranking only require $O(1)$ time for query operations. As shown in Figures 3(b), 4(b), and 5(b), **DAgH** achieves the highest **P@H=2** results on all the datasets. In particular, **P@H=2** of **DAgH** with 24 bits achieves the best performance. This shows that **DAgH** can learn more quality hash codes. Norouzi *et al.* [63] show that when generating relatively longer hash codes, the Hamming space will become sparse and few data points will fall within the Hamming ball with a radius of 2. This is why many learning-based hashing methods can achieve good image retrieval performances on short hash codes.

The other important indicators are Precision-Recall curves (**PR**) and Precision curves with a different Number of top returned samples (**P@N**). These results are shown in Figures 3(a), 4(a), 5(a) and Figures 3(c), 4(c), 5(c), respectively. We can observe that the performance of our proposed

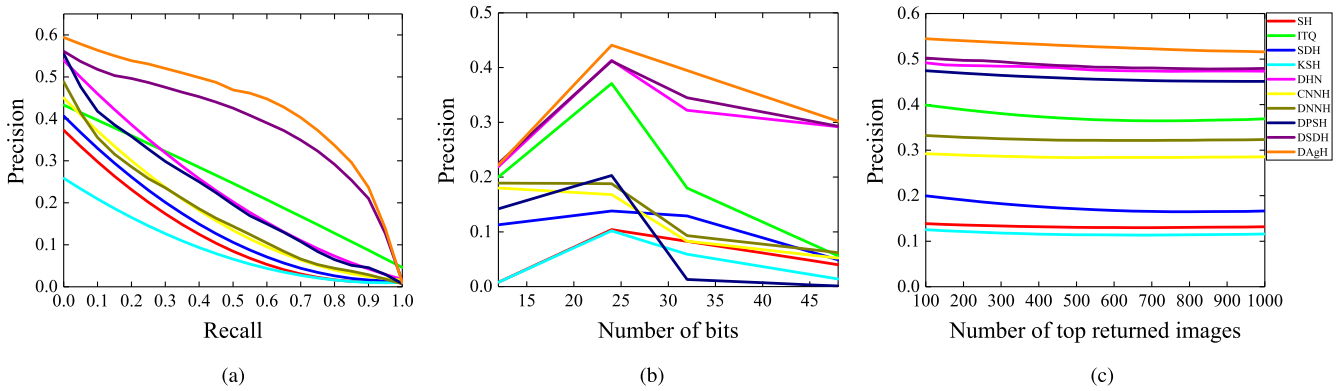


FIGURE 5. The results of DAgH and comparison methods on the ImageNet dataset under three evaluation metrics. (a) Precision-Recall curve @ 48 bits. (b) Precision within Hamming radius 2. (c) Precision curve w.r.t. top-*n* @ 48 bits.

model (**DAgH**) is better than the models to which it was compared. For example, using the proposed model, more semantic neighbors are retrieved, which is desirable in practical applications. In particular, **DAgH** achieves stable precision improvement at every recall level test and tests on the number of top images returned, which is very useful for real-world practical systems.

E. OTHER ANALYSIS

1) IMPACT OF THE FIRST HASHING NETWORK SELECTION

As shown in Figure 2, (the architecture of **DAgH**), we leverage a hashing network in the first stream framework to generate the attention-guided hash codes from the attention images. Intuitively, the performance of the hashing network could affect the quality of the attention-guided hash codes, i.e., the better the attention-guided hash codes is, the better the performance achieved by the second hashing network. To confirm this, we further design a new variant of **DAgH**, i.e., **DAgH-ResNet18**, which adopts ResNet-18 as the first hashing network, instead of AlexNet used in previous experiments. ResNet is a well-known convolutional neural network, and its performance in image processing is better than that of AlexNet. We carried out experiments on the NUS-WIDE dataset. The mAP results are shown in Table 2. **DAgH-AlexNet** implies that AlexNet was used in the first stream framework and ResNet-18 was used in **DAgH-ResNet18**. From table 2, the following observations were made:

1) **DAgH-ResNet18** outperforms **DAgH-AlexNet** in most cases except in the case of 48 bits. This proves that **DAgH** can obtain better results by using a first hashing network with better performance.
2) The performance gap between **DAgH-ResNet18** and **DAgH-AlexNet** was very small. This indicate that **DAgH** is not sensitive to attention-guided hash codes, this may be because the information of the attention hash codes is diluted when they guide the generation of new hash codes.

TABLE 2. Performance comparison of DAgH with different first hashing networks, i.e, AlexNet and ResNet18.

Method	12 bits	24 bits	32 bits	48 bits
DAgH-AlexNet	0.753	0.791	0.811	**0.825**
DAgH-ResNet18	**0.769**	**0.796**	**0.814**	0.823

2) IMPACT OF THE HYPER-PARAMETERS

In this subsection, we analyze the impact of the hyper-parameters, i.e., the value of the attention parameter ν and the margin parameter λ. The experiments are conducted on the NUS-WIDE dataset. The value of the attention penalty parameter ν is selected using values within the range 20 to 80 with a constant step-size of 10 and the margin parameter λ is using values within the range 0 to 0.5 with a constant step-size 0.05. Figure 6(a) shows that **DAgH** can achieve good performance on NUS-WIDE dataset within the range $40 \leq \nu < 60$. As shown in Figure 6(b), the model is sensitive to the value of the margin parameter λ and achieved good performance on NUS-WIDE dataset with $0.2 \leq \lambda \leq 0.35$. This is because according to (9), if the value of margin is small, the attention loss has a lower impact in punishing the attention network, and as the result, the attention image pair will be similar to the original image pair. If the value of margin is large, the attention loss will affect (10).

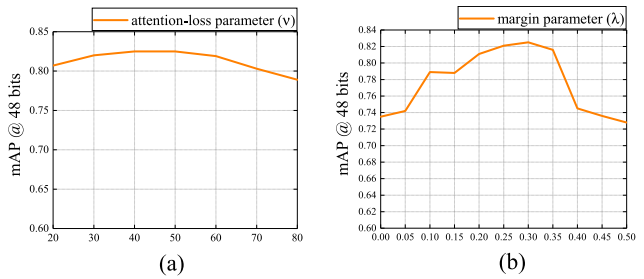


FIGURE 6. Influence of the hyper-parameters. (a) Value of weighting parameter. (b) Value of margin parameters.

3) ENCODING TIME

In practical retrieval systems, time efficiency for generating the hash codes for a new instance (image) is an important factor in the evaluation model. In this part, we compare the encoding time of the proposed **DAgH** method and other baseline hashing methods: **ITQ** [8], **SDH** [15], **KSH** [9], **DPSH** [52], **DHN** [16], **CNNH** [17], **DNNH** [33], and **DSDH** [34]. Since the input instances are originally raw images, for fair comparison, we take into consideration both the time cost for feature extraction and hashing encoding. We report both the feature extraction time efficiency for traditional hashing methods and the encoding cost for learning-based hashing methods on GPU and the hashing encoding time of traditional hashing methods on CPU. The encoding times (in microseconds, base 10) of involved hashing methods are presented in Figure 7 using a logarithmic scale on the CIFAR-10 dataset with 48 bits hash codes. From Figure 7, it can be seen that traditional hashing methods such as ITQ, and KSH, actually perform quite decently with encoding times faster than leaning-based hashing methods by an order of magnitude. However, traditional hashing methods require a separate process for feature extraction. When the full process of using a traditional method is put into consideration (feature extraction + traditional hashing method), the encoding time of the traditional methods is much worse than that of leaning-based hashing methods by an order of magnitude. The computing platform is equipped with an Intel 2× Intel E5-2600 CPU, 128G RAM, and a NVIDIA TITAN Xp 12G GPU. The encoding time basically depends on the adopted neural network model rather than the hashing method. Thus the time varies little with different lengths of hash codes.

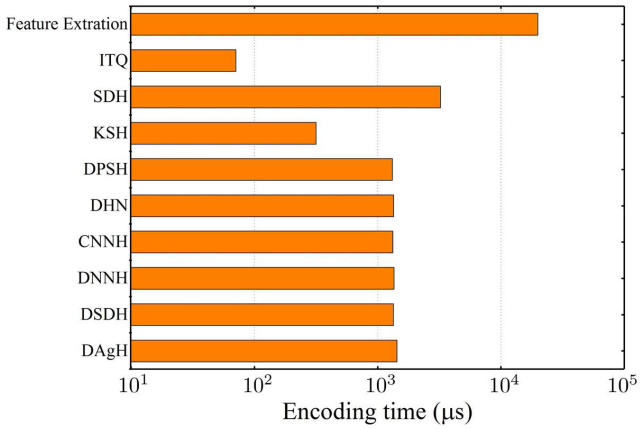


FIGURE 7. The encoding times to encode one new instance (image) of different hashing methods on CIFAR-10 dataset with 48 bits hash codes.

4) VISUALIZATION OF HASH CODES

Figure 8 shows the t-SNE visualization [64] of the hash codes learned by the proposed **DAgH** method and the best learning-based hashing baseline DSDH on the ImageNet dataset (we sample 10 categories for the case of visualization). We can observe that the hash codes generated by **DAgH** exhibit clear discriminative structures where the hash codes in different categories are well separated, while the hash codes generated by DSDH do not show such discriminative structures. The results verify that the hash codes learned through the proposed **DAgH** are more discriminative than those learned by DSDH, enabling more effective image retrieval.

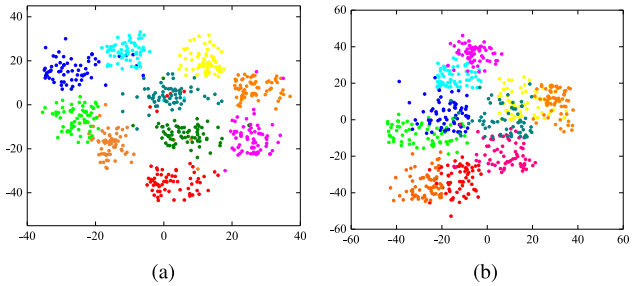


FIGURE 8. The t-SNE visualization of hash codes learned by DAgH and DSDH. (a) DAgH. (b) DSDH.

V. CONCLUSION

In this paper, we propose a novel attention-guided hashing method for image retrieval, named **DAgH**. To improve the quality of the generated hash codes, in other words, to address the high correlation problems of the generated hash codes, our method consists of two stream frameworks, which consist of an attention network and two hashing networks. The attention network can automatically mine the key region of an image and generate the attention images. The hashing networks are used to learn semantic-preserving hash codes. The first hashing network generates the attention-guided hash codes from the attention images using pairwise labels to learn the attention-guided hash codes. The second hashing network is then guided by the attention-guided hash codes to generate the final hash codes. On the choice of the hash activation function, the first stream framework uses a continuous ATanh activation function for training and the second stream framework uses a threshold function $sign(\cdot)$. Comprehensive experiments on the three benchmark image retrieval datasets demonstrate that the **DAgH** outperforms the state-of-the-art methods.

In the future, we plan to extend the self-hashing network to support image retrieval with relative similarity labels, i.e., condense the two stream framework into a single self-training network.

ACKNOWLEDGMENT

The authors would like to thank the associate editor and anonymous reviewers for their comments to improve the paper.

REFERENCES

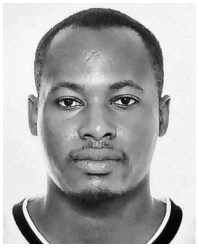
[1] Y. Cao, M. S. Long, J. M. Wang, and S. C. Liu, "Deep visual-semantic quantization for efficient image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 916–925.

[2] Z. K. Chen, F. M. Zhong, G. Y. Min, Y. L. Leng, and Y. M. Ying, "Supervised intra- and inter-modality similarity preserving hashing for cross-modal retrieval," *IEEE Access*, vol. 6, pp. 27796–27808, 2018.

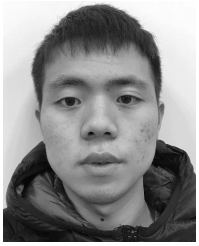
[3] Y. C. Guo, G. G. Ding, J. G. Han, and Y. Gao, "SitNet: Discrete similarity transfer network for zero-shot hashing," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, Aug. 2017, pp. 1767–1773.
[4] T. T. Yuan, W. H. Deng, and J. N. Hu, "Distortion minimization hashing," *IEEE Access*, vol. 5, pp. 23425–23435, 2017.
[5] F. Zheng, Y. Tang, and L. Shao, "Hetero-manifold regularisation for cross-modal hashing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1059–1071, May 2018.
[6] F. Zheng and L. Shao, "Learning cross-view binary identities for fast person re-identification," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, Jul. 2016, pp. 2399–2406.
[7] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proc. Int. Conf. Neural Inf. Process. (NIPS)*, Dec. 2008, pp. 1753–1760.
[8] 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.
[9] W. Liu, J. Wang, R. 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.
[10] J. Youn, J. Shim, and S. G. Lee, "Efficient data stream clustering with sliding windows based on locality-sensitive hashing," *IEEE Access*, vol. 6, pp. 63757–63776, 2018.
[11] M. Norouzi and D. J. Fleet, "Minimal loss hashing for compact binary codes," in *Proc. 8th Int. Conf. Mach. Learn.*, Jun. 2011, pp. 353–360.
[12] J. F. Wang, J. D. Wang, N. H. Yu, and S. P. Li, "Order preserving hashing for approximate nearest neighbor search," in *Proc. Conf. ACM Multimedia*, Jan. 2013, pp. 133–142.
[13] M. Norouzi, D. J. Fleet, and R. Salakhutdinov, "Hamming distance metric learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2012, pp. 1070–1078.
[14] R. Salakhutdinov and G. Hinton, "Semantic hashing," *Int. J. Approx. Reasoning*, vol. 50, no. 7, pp. 969–978, Jul. 2009.
[15] F. Shen, C. H. Shen, W. Liu, and H. T. Shen, "Supervised discrete hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 37–45.
[16] H. Zhu, M. S. Long, J. M. Wang, and Y. Cao, "Deep hashing network for efficient similarity retrieval," in *Proc. 13th Conf. Amer. Assoc. Artif. Intell.*, Feb. 2016, pp. 2415–2421.
[17] P. K. Xia, Y. Pan, H. J. Lai, C. Liu, and S. C. Yan, "Supervised hashing for image retrieval via image representation learning," in *Proc. Conf. Artif. Intell.*, Jul. 2014, pp. 2156–2162.
[18] J. Wu, Y. He, X. N. Guo, Y. J. Zhang, and N. Zhao, "Heterogeneous manifold ranking for image retrieval," *IEEE Access*, vol. 5, pp. 16871–16884, 2017.
[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. (NIPS)*, Dec. 2012, pp. 1106–1114.
[20] K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[21] J. Wang, T. Zhang, J. Song, N. Sebe, and H. T. Shen, "A survey on learning to hash," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 769–790, Apr. 2018.
[22] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of Massive Datasets*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2014.
[23] B. Kulis and K. Grauman, "Kernelized locality-sensitive hashing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1092–1104, Jun. 2012.
[24] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on P-stable distributions," in *Proc. 20th Annu. Symp. Comput. Geometry*, Jun. 2004, pp. 253–262.
[25] 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.
[26] Y. C. Gong and S. Lazebnik, "Iterative quantization: A procrustean approach to learning binary codes," in *Proc. CVPR*, Jun. 2011, pp. 817–824.
[27] R. Salakhutdinov and G. E. Hinton, "Learning a nonlinear embedding by preserving class neighbourhood structure," in *Proc. 11th Int. Conf. Artif. Intell. Statist.*, Mar. 2007, pp. 412–419.
[28] Y. Cao *et al.*, "Binary hashing for approximate nearest neighbor search on big data: A survey," *IEEE Access*, vol. 6, pp. 2039–2054, 2017.
[29] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2008, pp. 1753–1760.
[30] W. Liu, J. Wang, S. Kumar, and S. F. Chang, "Hashing with graphs," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, Jun. 2011, pp. 1–8.
[31] F. Zhao, Y. Z. Huang, L. Wang, and T. N. Tan, "Deep semantic ranking based hashing for multi-label image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1556–1564.
[32] H. M. Liu, R. Wang, S. Shan, and X. Chen, "Deep supervised hashing for fast image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2064–2072.
[33] H. J. Lai, Y. Pan, Y. Liu, and S. C. Yan, "Simultaneous feature learning and hash coding with deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3270–3278.
[34] Q. Li, Z. Sun, R. He, and T. Tan, "Deep supervised discrete hashing," in *Proc. Int. Conf. Neural Inf. Process. (NIPS)*, Dec. 2017, pp. 2479–2488.
[35] Y. C. Guo, X. Zhao, G. G. Ding, and J. G. Han, "On trivial solution and high correlation problems in deep supervised hashing," in *Proc. 32nd Conf. Amer. Assoc. Artif. Intell. (AAAI)*, Feb. 2018, pp. 2240–2247.
[36] L. Zheng, Y. J. Huang, H. C. Lu, and Y. Yang. (Jan. 2017). "Pose invariant embedding for deep person re-identification." [Online]. Available: https://arxiv.org/abs/1701.07732
[37] X. Liu, T. Xia, J. Wang, and Y. Q. Lin. (Mar. 2016). "Fully convolutional attention localization networks: Efficient attention localization for fine-grained recognition." [Online]. Available: https://arxiv.org/abs/1603.06765
[38] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang, "The application of two-level attention models in deep convolutional neural network for fine-grained image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 842–850.
[39] Y. M. Shen, L. Liu, L. Shao, and J. K. Song, "Deep binaries: Encoding semantic-rich cues for efficient textual-visual cross retrieval," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4117–4126.
[40] S. Q. Ren, K. M. He, R. B. Girshick, and J. Sun. (Jun. 2015). "Faster R-CNN: Towards real-time object detection with region proposal networks." [Online]. Available: https://arxiv.org/abs/1506.01497
[41] J. L. Bai *et al.*, "Deep progressive hashing for image retrieval," in *Proc. ACM Multimedia Conf.*, Oct. 2017, pp. 208–216.
[42] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Proc. Int. Conf. Neural Inf. Process. (NIPS)*, Dec. 2007, pp. 545–552.
[43] B. Zhao, X. Wu, J. S. Feng, Q. Peng, and S. C. Yan, "Diversified visual attention networks for fine-grained object classification," *IEEE Trans. Multimedia*, vol. 19, no. 6, pp. 1245–1256, Jun. 2017.
[44] J. Fu, J. Wang, Y. Rui, X. J. Wang, T. Mei, and H. Lu, "Image tag refinement with view-dependent concept representations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 8, pp. 1409–1422, Aug. 2015.
[45] J. Fu, Y. Wu, T. Mei, H. Wang, H. Lu, and Y. Rui, "Relaxing from vocabulary: Robust weakly-supervised deep learning for vocabulary-free image tagging," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1985–1993.
[46] S. Zagoruyko and N. Komodakis, "Wide Residual Networks," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Sep. 2016, pp. 1–12.
[47] J. Wang, J. Fu, Y. Xu, and T. Mei, "Beyond object recognition: Visual sentiment analysis with deep coupled adjective and noun neural networks," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, Aug. 2016, pp. 3484–3490.
[48] L. M. Zhao, X. Li, J. D. Wang, and Y. T. Zhuang. (Jul. 2017). "Deeply-learned part-aligned representations for person re-identification." [Online]. Available: https://arxiv.org/abs/1707.07256
[49] T. Y. Lin, A. R. Chowdhury, and S. Maji, "Bilinear CNN models for fine-grained visual recognition," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1449–1457.
[50] J. L. Fu, H. L. Zheng, and T. Mei, "Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4476–4484.
[51] S. Jin. (Jul. 2018). "Deep saliency hashing." [Online]. Available: https://arxiv.org/abs/1807.01459
[52] W. J. Li, S. Wang, and W. C. Kang, "Feature learning based deep supervised hashing with pairwise labels," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, Jul. 2016, pp. 1711–1717.
[53] X. Liu, T. Xia, J. Wang, Y. Yang, F. Zhou, and Y. Q. Lin. (Mar. 2016). "Fully convolutional attention networks for fine-grained recognition." [Online]. Available: https://arxiv.org/abs/1603.06765
[54] Z. Cao, Z. Sun, M. Long, J. Wang, and P. S. Yu, "Deep priority hashing," in *Proc. Conf. ACM Multimedia*, Oct. 2018, pp. 1653–1661.

[55] Z. Cao, M. Long, J. Wang, and P. S. Yu, "HashNet: Deep learning to hash by continuation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 5609–5618.
[56] X. L. Li, D. Hu, and F. P. Nie, "Deep binary reconstruction for cross-modal hashing," in *Proc. ACM Multimedia Conf.*, Nov. 2017, pp. 1398–1406.
[57] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," M.S. thesis, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2009.
[58] T. S. Chua, J. H. Tang, R. C. Hong, H. J. Li, Z. P. Luo, and Y. Zheng, "NUS-WIDE: A real-world web image database from national university of Singapore," in *Proc. Int. Conf. Image Video Retr. (CIVR)*, Jul. 2009, pp. 1–9.
[59] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[60] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. (May 2014). "Return of the devil in the details: Delving deep into convolutional nets." [Online]. Available: https://arxiv.org/abs/1405.3531
[61] J. Donahue *et al.*, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jun. 2014, pp. 647–655.
[62] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.
[63] M. Norouzi, A. Punjani, and D. Fleet, "Fast exact search in Hamming space with multi-index hashing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 6, pp. 1107–1119, Jun. 2014.
[64] L. van der Maaten and G. Hinton, "Visualizing high-dimensional data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

ZHAN YANG received the B.Sc. degree in environmental science from Nanjing Normal University and the M.Sc. degree in computer science and technology from Hainan University. He is currently pursuing the Ph.D. degree in computer science with Central South University, Changsha, China. His research interests include machine learning, the Internet of Things, and edge computing.



OSOLO IAN RAYMOND received the B.Tech. degree in electrical engineering from Nelson Mandela University, South Africa, and the M.Eng. degree in software engineering from Central South University, China, where he is currently pursuing the Ph.D. degree in computer science application and technology. His research interests include machine learning and embedded systems.



WUQING SUN received the B.Eng. degree in computer science and technology from Yangtze University, China, in 2017. He is currently pursuing the M.A.Eng. degree in computer technology with Central South University. His main research interests include machine learning and computer vision.

JUN LONG is currently a Professor with the School of Information Science and Engineering, Central South University, China, and also the Director of the Network Resources Management and Trust Evaluation Key Laboratory of Hunan Province, Central South University. His major research interests include machine learning and big data analysis.