IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 29, 2021, accepted February 16, 2021, date of publication February 24, 2021, date of current version March 4, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3061765*

# Pairwise Context Similarity for Image Retrieval System Using Variational Auto-Encoder

**HYEONGU YUN, YONGIL KIM, TAEGWAN KANG, AND KYOMIN JUNG, (Member, IEEE)**
Department of Electrical and Computer Engineering, Seoul National University, Seoul 08826, South Korea

Corresponding author: Kyomin Jung (kjung@snu.ac.kr)

This work was supported in part by Samsung Electronics Co., Ltd (IO201208-07852-01), in part by the Brain Korea 21 (BK21) FOUR Program of the Education and Research Program for Future Information and Communication Technology (ICT) Pioneers, Seoul National University, in 2021, and in part by the Automation and Systems Research Institute (ASRI), Seoul National University.

**ABSTRACT** Deep-learning-to-hash models have recently achieved several breakthroughs enabling a fast and efficient image retrieval system. As supervision for deep-learning-to-hash models, pairwise label similarity which considers two images to be identical if their labels are identical plays a crucial role. However, models using only pairwise label similarity cannot incorporate rich contextual information in images because pairwise label similarity solely depends on labels. In this paper, we initially address two major limitations of using the pairwise label similarity as only supervision for the deep-learning-to-hash model. Then, we propose a novel pairwise context similarity to alleviate those limitations. The proposed pairwise context similarity is computed on the latent space of a Variational Auto-Encoder which is trained in an unsupervised fashion that does not utilize any label information. Moreover we propose the strategy of an auxiliary loss for deep-learning-to-hash models that can easily be combined with previous losses using pairwise label similarity without deteriorating the retrieval quality. In our experiments on three standard benchmark datasets, our proposed method achieved high retrieval quality for image retrieval tasks while also showing advantages with regard to the addressed limitations. Also, we empirically prove that our proposed method acts as a proper regularization term during training so that our loss term therefore helps to mitigate overfitting and stabilizes the training curves.

**INDEX TERMS** Deep learning to hash, image feature hashing, supervised hashing, pairwise similarity, context similarity.

## I. INTRODUCTION

With image data overflowing on the web, deep-learning-to-hash models, which use a data driven approach, have been widely studied in an effort to achieve high quality outcomes during image retrieval tasks [6], [17], [26], [35], [43]. A deep-learning-to-hash model extracts a compact $K$-bit binary code $h_i$ of a given data point $x_i \in \mathbb{R}^D$ while preserving its contexts. Because computing the distance in the hash code space $dist(h_i, h_j)$ is more time-efficient than computing the distance in the original data space $dist(x_i, x_j)$, the extracted hash code $h_i$ is suitable for undertaking nearest neighbor searches or similarity ranking tasks [27] such as image retrieval tasks.

The main goal of a deep-learning-to-hash model is to generate hash codes from data, with a pair of hash codes considered to be similar if two paired given images are similar. To accomplish this goal, the model requires a similarity measure in the image space for training as a means of supervision. Indeed, there have been various works on similarity measures between images pairs including the Euclidean distance and its variants [2], [30], binary cross entropy, and exploiting low-level features [8], [18], [42]. Wang *et al.* [28], [29] introduced a pairwise *label* similarity which assumed that the contexts are the same if the labels of images are identical.

Given that Convolutional Neural Networks (CNN) architectures [11], [15] have shown outstanding performance using end-to-end fashion recently, pairwise *label* similarity is now frequently adopted in state-of-the-art deep-learning-to-hash models because it easily suits in end-to-end fashion. Several methods [4]–[6], [35], [43] have used pairwise *label* similarity as only supervision during training and have made breakthroughs in image feature hashing tasks.

Although the aforementioned methods achieved the state-of-the-art performance solely using pairwise *label* similarity,

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

in this work, we find critical disadvantages of using only pairwise *label* similarity which is not easily discernable on standard metrics such as the mean Average Precision (mAP). We point out two major limitations of using only pairwise *label* similarity; *scarce hash code diversity* and *low performance on mis-classified images*.

The *scarce hash code diversity* problem appears when the classifying performance of CNN is very high and the model is trained without any contextual information other than pairwise *label* similarity. Because the model is optimized to generate a similar pair of hash codes if labels are identical, it is eventually optimized to output the same hash codes for the same labeled images. Finally the feature hashing model with the *scarce hash code diversity* problem operates as an classifier and cannot rank images with the same labels. This problem differs from the overfitting problem which can be ascertained with learning curves.

The *low performance on mis-classified images* problem stems from the *scarce hash code diversity* problem. Often, the CNN part of the deep-learning-to-hash model mis-classifies a given image (e.g. it mis-classifies an airplane as a truck) because it cannot accomplish 100% test accuracy in the pre-training procedure. Given that the classifying CNN part plays a critical role in generating hash codes, the model generates a completely unrelated hash code (e.g. a hash code closer to truck images than airplane images) if it mis-classifies a given image. We observe a significant performance drop for mis-classified images even with state-of-the-arts models.

These two limitations can severely worsen the user-experience of image retrieval systems in real-world applications. For instance, a deep-learning-to-hash model with the *scarce hash code diversity* problem cannot properly distinguish images with the same label. Therefore, although retrieved images are likely to have the same label of a given query image, the model with the *scarce hash code diversity* problem cannot rank the suitability of retrieved images even if there are an irrelevant image. Also, a model with the *low performance on mis-classified images* problem shows a critical drawback which retrieves totally irrelevant images with different labels to a given query image if the query image is mis-classified by the CNN part of the model.

To overcome the limitations of deep-learning-to-hash models, we propose a novel pairwise *context* similarity measure capable of capturing the contextual information of images without using label information. Our proposed pairwise *context* similarity is a metric based on KL-divergence between the latent spaces of the Variational Auto Encoder (VAE) [13]. We especially focus on the VAE for two reasons: 1) As Arvanitidis *et al.* discussed in their earlier work [1], the VAE can be regarded as a surface model such that a distance can be defined on the latent space of the VAE. 2) Because VAE is a fully unsupervised model, we assume that the latent space captures the contexts of images without any additional information. With our proposed pairwise *context* similarity, we have empirically shown that a feature hashing model overcomes the disadvantages posed by using only pairwise *label* similarity through extensive experiments.

In the experiments, in order to verify that deep-learning-to-hash models only using pairwise label similarity encounter the *scarce hash code diversity* problem, we plot histograms of hamming distances between the hash code of query image and the hash codes of database images. The histograms of all baseline methods show that there are a large number of images with the same hash code for each query image. However, our proposed method with pairwise *context* similarity completely solves the *scarce hash code diversity* problem. We also show that our method can increase the average precision score on mis-classified images. A 64-bit model with our method shows improved performances on mis-classified images up to 56% compared to a 64-bit Hashnet model [6].

Our contribution are three fold; 1) We show that current deep-learning-to-hash methods using only pairwise label similarity have two major limitations that are often obscure when assessed with standard metrics such as mAP. These limitations can severely degrade the user experiences in real world applications. 2) We propose a novel pairwise distance measure for a pair of images which captures contextual information from the latent space of a VAE model. Our pairwise *context* similarity provides the similarity of the contexts between images, while an earlier pairwise label similarity only gives information about whether the images in a pair belong to the same class. 3) We present a novel feature hashing framework with our proposed pairwise similarity. Our feature hashing framework overcomes the limitations associated with pairwise label similarity. Also, our framework can enhance previous deep-learning-to-hash methods in addition to our proposed pairwise *context* similarity.

## II. RELATED WORKS

The most popular and widely used pairwise similarity as a supervision for image feature hashing model was Euclidean distance in image space [10], [16], [31], [40]. They attempted to capture contexts of images with the $l^2$-distance as pairwise similarity, but the retrieval performance was low. Binary Reconstructive Embedding (BRE) [16] minimized the difference between Hamming distance of two hash codes and Euclidean distance of two images. Topology Preserving Hashing (TPH) [40] was proposed to preserve the neighborhood ranking while preserving the data topology based on Euclidean distance. To enhance Euclidean distance, IMage Euclidean Distance (IMED) was introduced in [30] that utilized the spatial relationships of pixels. Also Euclidean distance in low-level feature space such as SIFT had also been used for image retrieval tasks [8], [18], [42]. These pairwise similarity measure can be interpreted as pairwise *context* similarity as it directly captures primitive contexts from a given pair of images. However, Our proposed measure is distinct in that we have fully utilized deep architecture to capture pairwise similarity by exploiting well-trained VAE.

There were also feature hashing methods without using pairwise similarity. Principal Component Hashing

(PCH) [21] was proposed as an unsupervised method using bucketing with principal direction to construct a hash table. Anchor Graph Hashing (AGH) [20] is proposed to use Anchor graphs to approximate the similarity neighbor graph. Semantic Hashing [23] used a deep generative model to generate hash codes and to reconstruct the input data. Our proposed method is partly close to [23] in the way that both methods exploit a deep generative models, but critically differs in that our method exploits the latent space of the deep generative to measure the pairwise *context* similarity between two data points.

Then methods using pairwise *label* similarity information at training stage were proposed. Kernel-Based Supervised Hashing (KSH) [19] was proposed as a feature hashing model using the pairwise *label* similarity to optimize the kernels for the hashing model. Minimum Loss Hashing (MLH) [22] was proposed to use a hinge-like pairwise loss term combining pairwise *label* similarity and Euclidean similarity. Supervised Discrete Hashing (SDH) [24] also used a pairwise hinge-like loss to optimize Hamming distance of hash codes across similar pairs of images.

As deep CNN architecture introduced, the deep-learning-to-hash methods made several breakthroughs with pairwise *label* similarity. Xia *et al.* [35] introduced Convolutional Neural Network Hashing (CNNH) as a feature hashing method with 2 stages; the hash code was learnt with pairwise label similarity in the first stage followed by the second stage where a deep convolutional hashing function was trained with the hash codes. Zhu *et al.* [43] proposed Deep Hashing Network (DHN) with a Bayesian framework to minimize pairwise cross entropy loss with pairwise *label* similarity and regularizing prior of hash codes with the bi-modal Laplacian prior to reduce the quantization error.

Several researches tried to reduce the quantization error which distort hash codes from the outputs of deep-learning-to-hash models. Deep Cauchy Hashing (DCH) [4] improved DHN by regularizing the prior distribution based on the Cauchy distribution, which enforced the model to concentrate similar data points within small Hamming distance ball. Cao *et al.* [6] proposed the continuation learning method called HashNet to reduce quantization error, which gradually reduced the smoothness of *tanh* activation function while increasing the multiplier $\beta$ for the activation function through the training. In [17], a simultaneous framework with CNN was proposed that the deep model was trained to get intermediate image features and the hash code was obtained from the intermediate features at the same time with the divide-and-encode method. Deep Quantization Network (DQN) [5] showed that using the product quantization approach improved the retrieval performance and reduced the quantization error at the same time. Su *et al.* [26] introduced a greedy algorithm which back-propagate the gradients by transmitting intactly in order to avoid the quantization error. Zhu *et al.* [44] exploited Gaussian distribution for posterior probability to constrain locality of hash codes since minimizing quantization error makes the features less discriminative.

On the other hand, Zhu *et al.* [44] also issued that current deep-learning-to-hash models suffers from less discriminative hash codes. Also, Wu *et al.* [34] pointed out the discrepancy between minimizing quantization error and discriminability of hash codes. They utilized a smooth projection function instead of quantization regularizer in order to preserve more context information. Yuan *et al.* [39] proposed a global similarity metric other than pairwise label similarity and introduced Central Similarity Quantization (CSQ) with an algorithm generating the center of hash codes in order to optimize the global similarity of the entire dataset. Xia *et al.* [36] defined pairwise multi-label supervision by leveraging multi-label and proposed a deep-learning-to-hash model that can hierarchically generate hash codes from images. Zhang *et al.* [41] utilized relative location relationship among multiple objects and defined a novel pairwise similarity of multi-label images in terms of location relationship.

Deep-learning-to-hash with unsupervised training were also studied recently. Wu *et al.* [33] proposed a transformation invariant binary feature descriptor which is trained via projecting geometrically transformed data into a joint binary space. Guo *et al.* [9] introduced a robust vector quantization algorithm which can be applied to $l_p$-norm similarity search. Yang *et al.* [38] regarded the distance of extracted features as semantic and noisy similarity. They proposed a distill procedure that infer true labels of data pairs automatically and adopt a Bayesian learning framework to generate hash codes. Wu *et al.* [32] proposed an unsupervised deep-learning-to-hash framework for large scale video retrieval task, reporting the state-of-the-arts performances.

Recent unsupervised deep-learning-to-hash models adopted the hidden space of the Auto Encoder. Shen *et al.* [25] split the hidden space of the Auto Encoder into binary space and continuous space, then used a code-driven adjacency graph to compute the distance in Hamming space. Xia *et al.* [37] regarded the hidden space of the stacked Auto Encoder as the mapping between image features and hash codes. Unlike their approaches, we utilize Variational Auto Encoder which transforms an image to a multivariate normal distribution so that the transformed vector represents not only the original image but also images similar to the original image. Our proposed pairwise similarity is computed based on KL-divergence between two multivariate normal distribution which differ from the Euclidean distance between two vectors from the hidden space of the Auto Encoder.

In order to make hash codes diverse and discriminative, [31], [36], [37] added a uncorrelation regularization constraint $\frac{1}{n}\sum_i h_i h_i^T = I$. The uncorrelation regularization term encourages the distribution of hash codes to be discriminative by forcefully pulling apart hash codes. Instead, our proposed pairwise similarity makes the distribution of hash codes to not only be diverse and discriminative but also follow the distribution of contextual similarity in the original image space.

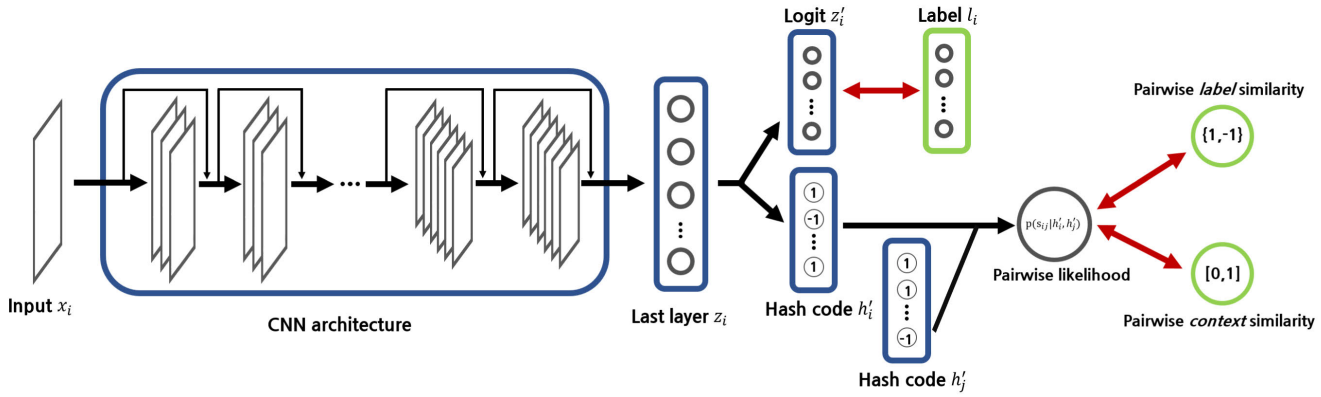


**FIGURE 1.** Overall architecture of the proposed feature hashing model in this paper. The parameters of CNN part $\theta_{cnn}$ are pre-trained with the cross entropy loss between the softmaxed logit $z'$ and the one-hot encoded label vector $l$. The parameters of hashing part $\theta_{hash}$ are trained as minimizing two kinds of pairwise losses;1)the cross entropy loss between the pairwise likelihood and the pairwise *label* similarity and 2)the KL-divergence between the pairwise likelihood and the pairwise *context* similarity. The second pairwise loss is proposed in this paper.

## III. FEATURE HASHING FRAMEWORK

In this section we will briefly review common parts of supervised feature hashing frameworks described in [4], [6], [43] using neural networks for image retrieval tasks.

The model $f : \mathcal{X} \to \mathcal{H}'$ extracts a pseudo $K$-bit hash code $h' \in [-1, 1]^K$ from a given input image $x$,

$$h' = f(x) \tag{1}$$

and a quantized $K$-bit binary hash code $h \in \{-1, 1\}^K$ is obtained by taking the sign of $h'$,

$$h = sign(h'). \tag{2}$$

For the image retrieval tasks, one can benefit from Hamming distance between two hash codes which is defined as,

$$dist_{hamming}(h_i, h_j) = \frac{1}{2}(K - \langle h_i, h_j \rangle), \tag{3}$$

where $h_i$ and $h_j$ denote the hash codes of $i$th and $j$th images and $\langle \cdot, \cdot \rangle$ is an inner product.

CNNs are especially preferred to build feature hashing architectures [4]–[6], [26], [35], [43] because of its ability to capture contexts of images. As shown in Figure 1, $z$, the second last layer of CNN, is passed to a hashing layer in order to output $h'$, where the hashing layer is a fully connected layer with *tanh* activation. For notational simplicity, we denote the CNN part as $f_{cnn}$ and the hashing layer as $f_{hash}$.

The training procedure of feature hashing model also can split down into two parts: 1)pre-training the CNN part and 2)training the hashing layer with simultaneously fine-tuning CNN part.

The parameters $\theta_{cnn}$ of the CNN part are pre-trained with cross entropy loss between the outputs of the softmax layer $z' \in (0, 1)^n$ and the one-hot encoded labels $l \in [0, 1]^n$ as usual classifier training with $n$ many classes. The likelihood $z'$ and the loss $L_{cnn}$ are defined as

$$z' = softmax(W_{logit} z + b_{logit}) \tag{4}$$

and

$$L_{cnn} = -\langle l, \log z' \rangle. \tag{5}$$

For training the parameters $\theta_{hash}$ of the hashing part, Zhu *et al.* [43] has proposed Bayesian framework with a given pairwise similarity $s_{ij}$. In [43], the Maximum a Posterior estimation $\log p(H'|S) \propto \log p(S|H')p(H')$ with the given pairwise similarity has been derived as follows,

$$\log p(S|H')p(H') = \sum_{s_{ij} \in S} \log p(s_{ij}|h'_i, h'_j)p(h'_i)p(h'_j), \tag{6}$$

where the pairwise *label* similarity $s_{ij}$ is

$$s_{ij} = \begin{cases} 1, & \text{if } x_i \text{ and } x_j \text{ has the same label} \\ 0, & \text{otherwise} \end{cases}. \tag{7}$$

The model defines the conditional likelihood of $s_{ij}$ as a pairwise logistic function and the loss function is defined as a pairwise cross entropy function as follows,

$$p(s_{ij}|h'_i, h'_j) = \begin{cases} \sigma(\langle h'_i, h'_j \rangle), & s_{ij} = 1 \\ 1 - \sigma(\langle h'_i, h'_j \rangle), & s_{ij} = 0 \end{cases} \tag{8}$$

and

$$L_{hash} = \log(1 + \exp\langle h'_i, h'_j \rangle) - s_{ij}\langle h'_i, h'_j \rangle. \tag{9}$$

Note that $L_{hash}$ uses the continuous $h'$ instead of the quantized hash code $h$. The quantization error problem can be reduced by regularizing the prior $p(h')$ with the bimodal Laplacian distribution [43], the Cauchy distribution [4], or the bimodal Gaussian distribution [3]. During training $\theta_{hash}$ with $L_{hash}$, the pre-trained parameters $\theta cnn$ can also be fine-tuned with adjusted learning rates.

## IV. LIMITATIONS OF PAIRWISE LABEL SIMILARITY

Although using the pairwise *label* similarity has achieved high mean average precision (mAP) score with supervised feature hashing architectures, there are two main limitations of using only pairwise *label* similarity.

### A. SCARCE HASH CODE DIVERSITY

The main limitation is that the model generates only few kinds of hash codes. In our experiments with CIFAR-10 dataset [14] and $K = 48$ bits, we find that the hash codes of 54,000 images resulting from the well-trained Hashnet have fallen into only 18,816 different hash codes. Despite the fact that there are total $2^{48}$ possible codes which is enough to generate 54,000 different kinds of hash codes, using only the pairwise *label* similarity enforces the model to generate the same code if the labels of images are the same. Although this tendency is positive to get a high mAP score, it causes a critical drawback in real world user experiences. As shown in Figure 2, well-trained Hashnet cannot tell how far retrieved images are from the query image since their Hamming distances are all 0s, even the Hamming distance from the query image and the ostrich image. The numbers below each of the images are the Hamming distances resulted from our methods indicating that our method can separate the ostrich image from the others. In the worst case, 4,007 out of 5,438 *car* images have fallen into the same hash code by the well-trained Hashnet.

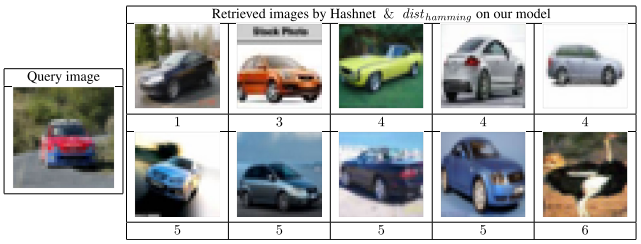


**FIGURE 2. 10 retrieved images for the given query image from the well-trained Hashnet. All of these 10 images are randomly sampled from database images with $dist_{hamming} = 0$. The numbers below each of the images are the Hamming distance outputs from our model.**

### B. LOW PERFORMANCE ON MIS-CLASSIFIED IMAGES

We also find that the precision has significantly dropped down when the pre-trained CNN mis-classifies the query image. The feature hashing model with only using the pairwise *label* similarity totally depends on the extracted representation $z$ of the CNN and the pairwise *label* similarity, hence the model often gets confused when the pre-trained CNN mis-classifies the input image. As shown in Figure 3, when an airplane image as the query image is an input to the Hashnet where the CNN mis-classifies it as a truck, the nearest 100 images found are mostly images of trucks and only 26 images are images of airplanes. With our pairwise *context* similarity, the model with the same pre-trained CNN is optimized with additional information about how far the context between two images and the model is able to find 54 airplanes in the nearest 100 images.

## V. PAIRWISE CONTEXT SIMILARITY

To alleviate these problems above, we propose a novel feature hashing framework with a pairwise *context* similarity which is learnt with an unsupervised fashion. Our pairwise *context* similarity is not involved with the labels and solely depends

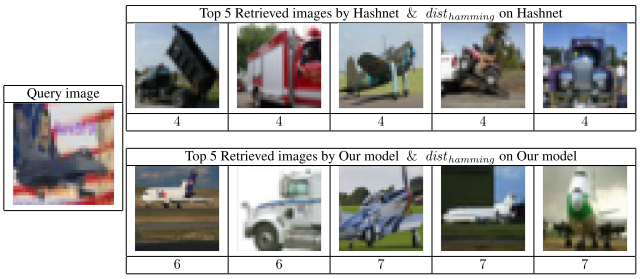


**FIGURE 3. Top 5 retrieved images results for the given query image from the well-trained Hashnet(top) and our proposed model(bottom). When the shared CNN part has mis-classified the query image(as an image of a truck), the retrieval performance of the Hashnet is very low. The bottom row indicates that training with our proposed pairwise context similarity can alleviate this problem. The numbers below each of images are the Hamming distances between the retrieved image and the query image.**

on the contexts of two images, hence our method relaxes the limitations raised by using only the pairwise *label* similarity. Also the loss term derived with pairwise *context* similarity can be combined with $L_{hash}$ at Equation 9 as taking benefits of all previous works while it does not harm the mAP performance.

### A. LATENT SPACE OF VAE

Variational Auto Encoder (VAE) [13] encodes an image into $d$-dimensional latent variables and decodes the latent variables to generate an output image. Kingma *et al.* have introduced the reparametrization trick [13] which encodes the latent variables into a mean vector $\mu$ and a diagonal covariance matrix $\Sigma$ of a multivariate Gaussian distribution. With reparametrization trick, an input of the decoder is sampled from the multivariate Gaussian distribution.

We focus on these two vectors, $\mu$ and $\Sigma$, as a compressed representation of an image. In a well-trained VAE model, the encoder of the VAE translates an image into a multivariate Gaussian distribution with the center vector $\mu$ and the diagonal of covariance matrix $\Sigma$. Then, the decoder of the VAE is able to generate similar images from vectors which are sampled from the multivariate Gaussian distribution. Therefore, if the performance of the VAE model is reliable, a multivariate Gaussian distribution with $\mu$ and $\Sigma$ is a suitable representation of an image. We use these vectors to calculate the similarity between images without any supervision.

In order to calculate the similarity between $(\mu_i, \Sigma_i)$ and $(\mu_j, \Sigma_j)$, we set the KL-divergence between two multivariate Gaussian distributions as the contextual similarity between given two images $x_i$ and $x_j$. The KL-divergence between two multivariate Gaussian distributions is defined as follows,

$$D_{KL}(\mathcal{N}_i||\mathcal{N}_j) = \frac{1}{2}(\log\frac{|\Sigma_j|}{|\Sigma_i|} - d + tr\{\Sigma_j^{-1}\Sigma_i\} + (\mu_j - \mu_i)^T\Sigma_j^{-1}(\mu_j - \mu_i)), \quad (10)$$

where $\mu_i$ and $\Sigma_i$ are outputs of the encoder with a given image $x_i$,

$$\left[\mu_i, \Sigma_i\right] = encoder(x_i), \tag{11}$$

and $\mathcal{N}_i(\mu_i, \Sigma_i)$ is a $d$-dimensional multivariate Gaussian distribution of an $i$-th image with a diagonal covariance matrix.

Although KL-divergence is asymmetric, it still gives us a relative and comparable measure between two images. As shown in Figure 4, the closer KL-divergence between two distributions is to 0, the more image is likely similar to the query image in human visual perception. Feature hashing models with using only pairwise *label* similarity cannot sort these sampled images since the *label* similarities are all 1s because they all have the same label, yet our proposed measurement can sort them by the relative measurements from the query image. We will simply call the KL divergence between two latent vectors of VAE as pairwise *context* similarity in the rest of this paper.

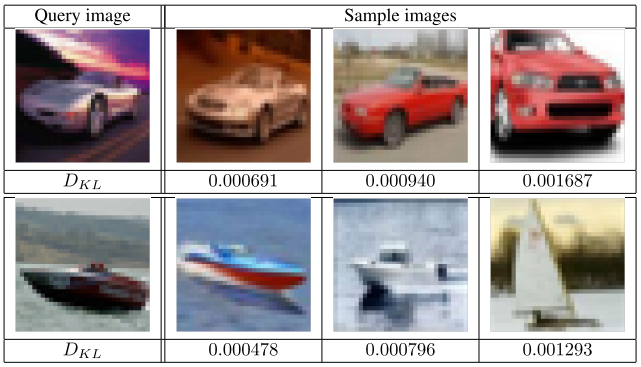


**FIGURE 4.** KL divergences $D_{KL}(\mathcal{N}_{sample}||\mathcal{N}_{query})$ with the query image and sampled images having the same labels.

### B. FEATURE HASHING WITH THE SIMILARITY MEASURING MODEL

Since the pairwise *context* similarity is defined as KL divergence, it can be easily combined with the existing pairwise *label* similarity loss. Firstly, we take the exponential of negative pairwise *context* similarity to make it a pseudo probability distribution, as follows,

$$q(s_{ij} = 1|h'_i, h'_j) = \exp\left(-\nu * D_{KL}(\mathcal{N}_i||\mathcal{N}_j)\right), \tag{12}$$

where $\nu$ is a hyperparameter.

Then we can minimize another KL divergence from $q$ to $p$, $D_{KL}(p(s_{ij} = 1|h'_i, h'_j)|q(s_{ij} = 1|h'_i, h'_j))$, in training to encourage the pairwise output likelihood to follow the pairwise *context* similarity distribution. In other words, the model will be optimized to the way that is more likely to generate a pair of similar hash codes when the pairwise *context* similarity is close to 0. The loss term with pairwise *context* similarity can be written as follows,

$$\begin{aligned} L_{context} &= D_{KL}(p(s_{ij} = 1|h'_i, h'_j)|q(s_{ij} = 1|h'_i, h'_j)) \\ &= -\sum \sigma(\langle h'_i, h'_j\rangle) \log \frac{\exp\left(-\nu * D_{KL}(\mathcal{N}_i||\mathcal{N}_j)\right)}{\sigma(\langle h'_i, h'_j\rangle)}. \end{aligned} \tag{13}$$

The final loss for training can be organized as a weighted sum of $L_{hash}$ and $L_{context}$ with a scaling hyperparameter $\lambda$, as follows,

$$L = (1 - \lambda)L_{hash} + \lambda L_{context}. \tag{14}$$

Furthermore, the prior regularizing terms that were introduced in [3], [4], [43] can be added to the final loss in order to reduce the quantization error.

Also, note that $L_{context}$ can be interpreted as a regularization term to prevent the overfitting problem. We have observed that, in some experiments, the model trained with our proposed loss has achieved a higher mAP score than the model without it.

Our proposed pairwise *context* similarity differs the multi-label semantic similarity proposed by Xia *et al.* [36] or the pairwise location similarity proposed by Zhang *et al.* [41] in that our proposed pairwise *context* similarity entirely depend on the latent space of VAE trained without any label supervision while the multi-label semantic similarity [36] is based on the label co-occurrence and the pairwise location similarity [41] is based on a local graph structure with the multi-label semantic similarity.

## VI. EXPERIMENTS

To show the strengths of our model, we compare our method with other baseline methods in standard deep-learning-to-hash metrics; mean Average Precision(mAP), Precision-Recall Curves, and Precision-Hamming distance curves. Then, by analyzing histogram of Hamming distance between hash codes and evaluating Average Precision on mis-classifying images, we prove that our model well reduces the two limitations we have addressed. All experiments are conducted on three datasets; CIFAR-10 [14], NUS-WIDE [7], and MIRFLICKR25k [12].

### A. DATASET

We follow all dataset set-ups as in [43].

- **CIFAR-10** [14]: CIFAR-10 consists of 60,000 images with 10 categories. Each image has 3 color channels and the size of 32 × 32. We resize each image to the size of 256 × 256 and center-crop it to the size of 224 × 224. We randomly sample 5,000 images as training images, 1,000 images as query images, and use remaining 54,000 images as database for the evaluation.
- **NUS-WIDE** [7]: NUS-WIDE consists of 269,648 multi-labeled images with 81 concepts collected on the web. We use a subset of 162,336 images labeled with 21 most frequent concepts. Among 162,336 images, we sample 10,500 images as training images, 2100 images as query images, and remaining 149,736 images as database. Each image is cropped to center and resized to the size of 224 × 224.
- **MIRFLICKR-25k** [12]: MIRFLICKR-25k consists of 25,000 images downloaded from the social photography site Flickr. The dataset is split randomly into 5,000 training images, 1,000 query images, and

**TABLE 1. Mean Average Precision (mAP) of Hamming distance for NUS-WIDE, CIFAR-10, and MirFlickr dataset. Our method shows comparable performances to baseline methods, which shows that our method does not harm mAP score.**

| Method | NUS-WIDE | | | | CIFAR-10 | | | | MIRFLICKR25k | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 16 bits | 32 bits | 48 bits | 64 bits | 16 bits | 32 bits | 48 bits | 64 bits | 16 bits | 32 bits | 48 bits | 64 bits |
| DHN | 0.790 | 0.805 | 0.815 | 0.817 | 0.800 | 0.797 | 0.799 | 0.804 | 0.755 | 0.765 | 0.768 | 0.775 |
| DCH | 0.754 | 0.755 | 0.756 | 0.751 | 0.755 | 0.777 | 0.783 | 0.783 | 0.749 | 0.760 | 0.757 | 0.758 |
| Hashnet | 0.714 | 0.786 | 0.805 | 0.819 | 0.549 | 0.789 | 0.809 | 0.832 | 0.745 | 0.763 | 0.774 | 0.777 |
| DSHSD | 0.676 | 0.774 | 0.777 | 0.784 | 0.758 | 0.761 | 0.784 | 0.783 | 0.745 | 0.762 | 0.768 | 0.770 |
| CSQ | 0.759 | 0.790 | - | 0.806 | 0.767 | 0.784 | - | 0.762 | 0.671 | 0.679 | - | 0.686 |
| GreedyHash | 0.735 | 0.761 | 0.778 | 0.782 | 0.754 | 0.778 | 0.795 | 0.810 | 0.681 | 0.680 | 0.700 | 0.707 |
| Our method | 0.782 | 0.804 | 0.816 | 0.820 | 0.791 | 0.798 | 0.805 | 0.799 | 0.756 | 0.766 | 0.770 | 0.776 |

20,000 database images. We also crop to center and resize each image to the size of 224 × 224.

### B. MODEL DETAILS

- **Shared CNN structure**: We use Alexnet [15] structure with residual connection for our pre-trained CNN part. Alexnet architecture consists of five convolution layers with pooling layers. To generate hash codes from image, Alexnet is follow by two fully connected layers with 4096 dimension and an output layer. For fair comparison, we share the weights of the pre-trained CNN for all comparing methods as the initial weights.
- **VAE**: To calculate the pairwise *context* similarity, we also implement a VAE structure. We use 256 dimension for the latent space of VAE. We train our VAE 50,000 steps with 128 images per a mini-batch. Images in Figure 5 are examples of generated outputs from VAE, showing the performance of our trained VAE.
- **Baseline models**: We compare mAP performance of our method along with recently introduced six deep-learning-to-hash models; DHN [43], DCH [4], HashNet [6], DSHSD [34], CSQ [39], and GreedyHash [26]. DHN and DCH decrease the quantization error by exploiting bi-modal Laplacian distribution and Cauchy distribution respectively as a prior for generating hash codes. HashNet and GreedyHash also reduces the quantization error with a continuation learning method and a greedy algorithm respectively. DSHSD use a smooth projection function to preserve context information. CSQ generates a center of hash codes and optimize the center with the global similarity of entire dataset. CSQ cannot be applied in 48-bit experiments because the algorithm to generate the center of hash codes requires the hash codes of $2^K$-bit. However, all six baseline methods depend on pairwise label similarity as a supervision.
- **Other hyperparameters**: For all experiments, we train models up to 75 epochs with early stopping. We set $\lambda_q = 0.1$ for the scaling parameter in **DHN** and $\gamma = 100$ for the scale parameter of the Cauchy distribution in **DCH**. We also train **Hashnet** with $\beta = 1$ at first and gradually increase to $\beta = 10$ as continuation learning method proposed in [6]. For scale parameter $\lambda$ and $\alpha$ in DSHSD and CSQ respectively, we set $\alpha = 0.05$ and $\lambda_1 = 0.0001$ in our experiments. We use $p = 3$ and $\alpha = 0.1$ as hyperparameters in GreedyHash Our proposed model is trained with loss term at Equation 14 with $\lambda = 0.1$ and $\nu = 0.01$.

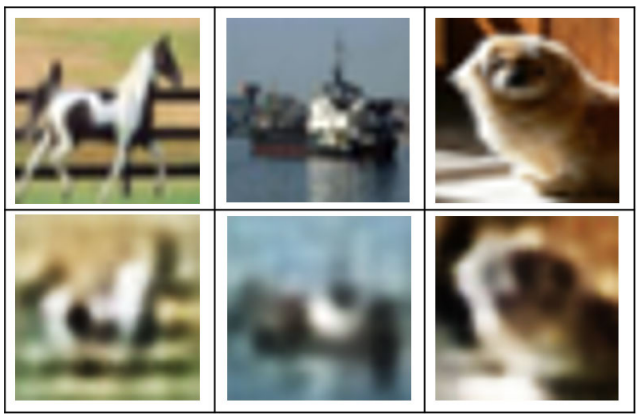

**FIGURE 5. Generated output images(bottom) for given images(top) shows the performance of our trained VAE.**

### C. ANALYSES WITH STANDARD METRICS

The mAP scores are reported in the Table 1. Our method achieves comparable performance with other state-of-the-arts models, and even shows the best results on some experiments. On NUS-WIDE dataset, our method reports the highest mAP scores for 48 bits and 64 bits hash codes, and it also shows good performance to be compared against the other two cases. In the case of CIFAR-10 dataset, our method shows the best performance for 32 bits hash code, and the results of different bit hash codes show similar performance to other state-of-the-art methods. Finally, on the MIRFLICKR25k dataset, ours shows the best performance in all hash codes except 64 bits hash code and especially outperforms DHN for all hash code cases. This is an unexpected side-benefit because our method does not accurately optimize mAP scores that depend on the label similarity. It is natural to have a strong point in mAP with a model learned with pairwise label similarity only, since mAP is measured with the labels of retrieved images. However, our proposed method does not lag behind at all and rather performs better in mAP even though it is trained with pairwise *context* similarity. This allows us to analyze that learning via pairwise *context* similarity is not at all disturbing in terms of performance measurement of mAPs.

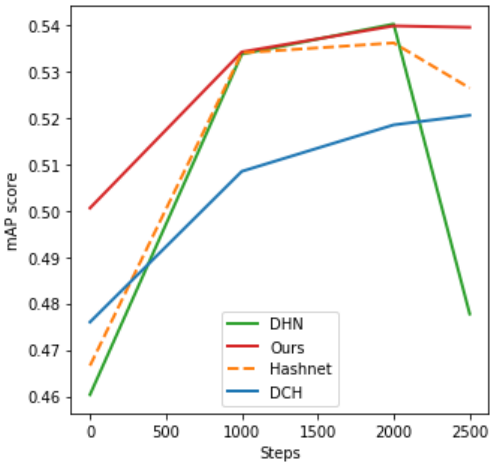


**FIGURE 6.** Validation curves shows the mAP values during training on NUS-WIDE dataset. Our methods prevent the overfitting indicating that it can be regarded as a proper regularization method.

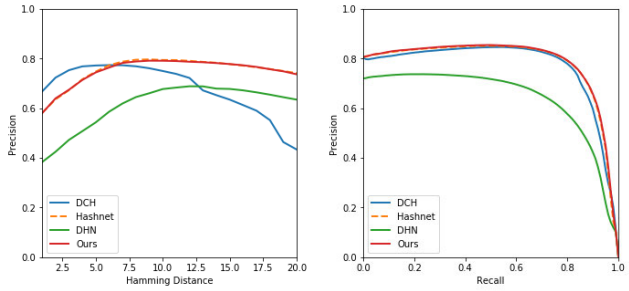


**FIGURE 7.** Precision-Hamming radius curves @ 64 (left) and Precision-recall curves @ 64 bits (right) of our method and other comparison methods on the CIFAR-10 dataset.

Figure 6 shows that our proposed loss term works as a proper regularization term that stabilizes the training curve and relaxes the over-fitting problem. As training step increases, the mAP validation scores of other baseline methods drop dramatically indicating that those models all suffers from overfitting problem, whereas the validation score of our proposed method does not decrease. Therefore, it can be interpreted that adopting our auxiliary loss prevents over-fitting problems and helps the model to converge smoothly.

Figure 7 shows the similar results. This figure shows the precision within the Hamming Distance at 64 bits hash code and precision-recall curves for each baseline models at 64 bits hash code. Our proposed model shows the best performance across the entire Hamming Distance, shown in the figure on the left. Although DCH scores the best precision with Hamming distance 2 and 5 (P@H$\leq$2, P@H$\leq$5), the precision of the model significantly drops down with Hamming distance radius larger than 7. Also, our method has larger area under precision-recall curve than any other methods. This shows that our model is generally the most reliable and robust model.

### D. ANALYSES ON LIMITATIONS OF PAIRWISE LABEL SIMILARITY

The forthcoming analyses reveal that our method is a remedy for the two limitations of using only pairwise *label* similarity; **Scarce hash code diversity** and **Low performance on mis-classified images**.

Figure 8 shows that our method solves the *scarce hash code diversity* problem. Each histogram in Figure 8 is a histogram of Hamming distances between query and output hash codes from CIFAR-10 experiments. DHN, DCH, and Hashnet all suffered from *scarce hash code diversity* problem represented by a large number at Hamming distance of zero (left-most bar of each histogram). The leftmost bar of each histogram shows an average number of images with $dist_{hamming}(h_i, h_j) = 0$ for each query images, which imply the average number of images that the deep-learning-to-hash model interprets as the same image to the query image. Note that for DCH, there are 2,062 images with $dist_{hamming}(h_i, h_j) = 0$ and 1,536 images for Hashnet. The large number of image pairs with $dist_{hamming}(h_i, h_j) = 0$ means that many images are concentrated in the same hash code, which is a solid evidence of the scarcity problem in deep-learning-to-hash models. However, our proposed model trained with pairwise *context* similarity has well diversified the distribution of output hash codes. There is only one image $j$ with $dist_{hamming}(h_i, h_j) = 0$ for each query image $i$ with our method, which is the query image itself.

In Figure 9, we also plotted $\exp(-\nu * D_{KL}(\mathcal{N}_i||\mathcal{N}_j))$ at Equation 12 to show that a shape of the KL divergence between two multivariate Gaussian is a proper pairwise *context* similarity. Equation 12 with proper $\nu$ can be regarded as a probability that $i$-th image $x_i$ and $j$-th image $x_j$ are similar. Because minimizing our context similarity loss term forces the desired hash code distribution to resemble the distribution of Figure 9, we can see that the KL-divergence between the latent spaces of VAE is very helpful in solving the scarcity problem by comparing the shapes of Figure 8 to the shape of Figure 9.

We conduct another experiment to prove that our method can increase the performance on mis-classified images. We collect mis-classified images by the shared CNN structure from CIFAR-10 (i.e. a "plane" labeled image which Alexnet mis-classifies as a "truck") and report the average precision of $dist_{hamming} < 2$ in Table 2. We clearly see the low performance on mis-classified images as the average precision dropped down severely for all methods. Nevertheless, compared to other baseline methods, our proposed method with pairwise *context* similarity scores the highest average precision for all models. Hashnet [6] shows the least performance drop for 32 bits, but it suffers critically for 48 bits and 64 bits model. These results indicate that our method is effective for reducing the *low performance problem on mis-classified images*. We believe that the increased performance from mis-classified images also affects the total mAP score.

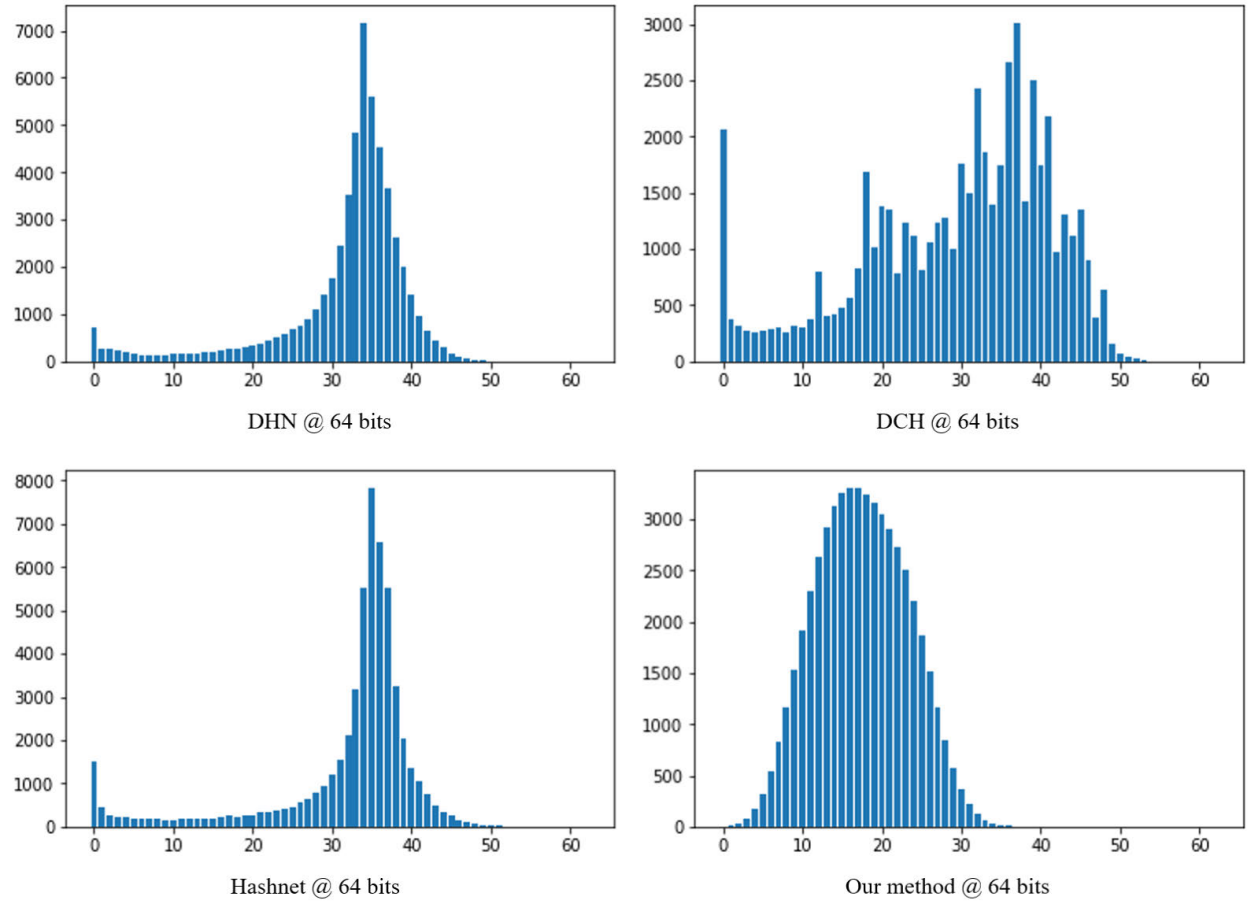


**FIGURE 8.** Histograms of Hamming distances between the query and output hash codes(the value is averaged by the number of query images) of our method and comparing 3 methods in the CIFAR-10 experiment. All three comparing methods have a biased number at Hamming distance 0, indicating that *scarce hash code diversity* problem can be posed. Note that there are average of 2,062 images with Hamming distance 0 for DCH and 1,536 images for Hashnet while the histogram of our method have a smooth distribution over Hamming distances.

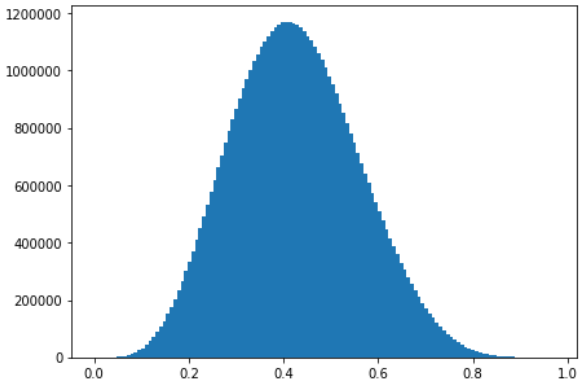

**FIGURE 9.** Histograms of $\exp(-1000 * D_{KL}(\mathcal{N}_i || \mathcal{N}_j))$ at Equation 12 for all $(i, j)$ pairs in CIFAR-10 database.

**TABLE 2.** Average Precision of Hamming radius < 2 on mis-classified images for CIFAR-10 dataset.

| Method | 32 bits | 48 bits | 64 bits |
|---|---|---|---|
| DHN | 0.091 | 0.030 | 0.069 |
| DCH | 0.145 | 0.117 | 0.127 |
| Hashnet | 0.177 | 0.064 | 0.064 |
| DSHSD | 0.103 | 0.089 | 0.117 |
| CSQ | 0.158 | - | 0.149 |
| GreedyHash | 0.132 | 0.093 | 0.127 |
| Our method | **0.190** | **0.144** | **0.164** |

The trade-off of our proposed method is a time efficiency in training time. Because our method requires the encoding procedure of the VAE and calculating $L_{context}$ for each training step, the training time is increased compared to the training time of DHN. For example, in CIFAR-10 experiments, the average training time per epoch of HashNet is 9.47 seconds, while the average training time per epoch for the same HashNet with our method takes 26.23 seconds. However, the test time remains the same because the deep-learning-to-hash model with our method does not need to compute $L_{context}$ when it generates hash codes from the data.

## VII. CONCLUSION

We address two possible limitations posed by using only pairwise *label* similarity in feature hashing models for image retrieval task. *Scarce hash code diversity* and *low performance on mis-classified images* do harm to the retrieval quality. Then we also introduced a method with a novel auxiliary

loss using pairwise *context* similarity which derived from the latent space of VAE to alleviate those problems. Our proposed pairwise *context* similarity can be trained in an unsupervised fashion and also it can easily combine with existing pairwise losses for feature hashing models. Empirical evidences show that our proposed method can solve the *Scarce hash code diversity* without deteriorating the standard metrics. Also, we find that our proposed loss acts as a proper regularization term through our experiments. We believe that our method is easily adopted in many real-world applications and improves the user-experience by alleviating the aforementioned limitations of current deep-learning-to-hash models.

## REFERENCES

[1] G. Arvanitidis, L. Kai Hansen, and S. Hauberg, "Latent space oddity: On the curvature of deep generative models," 2017, *arXiv:1710.11379*. [Online]. Available: http://arxiv.org/abs/1710.11379
[2] M. Broilo and F. G. B. De Natale, "A stochastic approach to image retrieval using relevance feedback and particle swarm optimization," *IEEE Trans. Multimedia*, vol. 12, no. 4, pp. 267–277, Jun. 2010.
[3] Y. Cao, B. Liu, M. Long, and J. Wang, "HashGAN: Deep learning to hash with pair conditional wasserstein GAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1287–1296.
[4] Y. Cao, M. Long, B. Liu, and J. Wang, "Deep cauchy hashing for Hamming space retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1229–1237.
[5] 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. 3457–3463.
[6] Z. Cao, M. Long, J. Wang, and P. S. Yu, "HashNet: Deep learning to hash by continuation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5608–5617.
[7] 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.*, Santorini, Greece, 2009, pp. 1–9.
[8] E. Delponte, F. Isgrá, F. Odone, and A. Verri, "SVD-matching using SIFT features," *Graph. Models*, vol. 68, nos. 5–6, pp. 415–431, Sep. 2006.
[9] Y. Guo, G. Ding, and J. Han, "Robust quantization for general similarity search," *IEEE Trans. Image Process.*, vol. 27, no. 2, pp. 949–963, Feb. 2018.
[10] J. He, S.-F. Chang, R. Radhakrishnan, and C. Bauer, "Compact hashing with joint optimization of search accuracy and time," in *Proc. CVPR*, Jun. 2011, pp. 753–760.
[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[12] M. J. Huiskes and M. S. Lew, "The MIR flickr retrieval evaluation," in *Proc. 1st ACM Int. Conf. Multimedia Inf. Retr.*, 2008, pp. 39–43.
[13] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*. [Online]. Available: http://arxiv.org/abs/1312.6114
[14] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009.
[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[16] B. Kulis and T. Darrell, "Learning to hash with binary reconstructive embeddings," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1042–1050.
[17] 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. (CVPR)*, Jun. 2015, pp. 3270–3278.
[18] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman, "Sift flow: Dense correspondence across different scenes," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2008, pp. 28–42.
[19] 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.
[20] W. Liu, J. Wang, S. Kumar, and S.-F. Chang, "Hashing with graphs," in *Proc. 28th Int. Conf. Int. Conf. Mach. Learn. (ICML)*, Jun. 2011, pp. 1–8.
[21] Y. Matsushita and T. Wada, "Principal component hashing: An accelerated approximate nearest neighbor search," in *Proc. Pacific-Rim Symp. Image Video Technol.* Berlin, Germany: Springer, 2009, pp. 374–385.
[22] 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.
[23] R. Salakhutdinov and G. Hinton, "Semantic hashing," *Int. J. Approx. Reasoning*, vol. 50, no. 7, pp. 969–978, Jul. 2009.
[24] F. Shen, C. Shen, W. Liu, and H. T. Shen, "Supervised discrete hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 37–45.
[25] Y. Shen, J. Qin, J. Chen, M. Yu, L. Liu, F. Zhu, F. Shen, and L. Shao, "Auto-encoding twin-bottleneck hashing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2818–2827.
[26] S. Su, C. Zhang, K. Han, and Y. Tian, "Greedy hash: Towards fast optimization for accurate hash coding in CNN," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 806–815.
[27] J. Wang, H. Tao Shen, J. Song, and J. Ji, "Hashing for similarity search: A survey," 2014, *arXiv:1408.2927*. [Online]. Available: http://arxiv.org/abs/1408.2927
[28] J. Wang, S. Kumar, and S.-F. Chang, "Sequential projection learning for hashing with compact codes," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, Jun. 2010, pp. 1127–1134.
[29] J. Wang, S. Kumar, and S.-F. Chang, "Semi-supervised hashing for large-scale search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2393–2406, Dec. 2012.
[30] L. Wang, Y. Zhang, and J. Feng, "On the Euclidean distance of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1334–1339, Aug. 2005.
[31] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1753–1760.
[32] G. Wu, J. Han, Y. Guo, L. Liu, G. Ding, Q. Ni, and L. Shao, "Unsupervised deep video hashing via balanced code for large-scale video retrieval," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1993–2007, Apr. 2019.
[33] G. Wu, Z. Lin, G. Ding, Q. Ni, and J. Han, "On aggregation of unsupervised deep binary descriptor with weak bits," *IEEE Trans. Image Process.*, vol. 29, pp. 9266–9278, Sep. 2020.
[34] L. Wu, H. Ling, P. Li, J. Chen, Y. Fang, and F. Zhou, "Deep supervised hashing based on stable distribution," *IEEE Access*, vol. 7, pp. 36489–36499, 2019.
[35] 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. 2156–2162.
[36] Z. Xia, X. Feng, J. Lin, and A. Hadid, "Deep convolutional hashing using pairwise multi-label supervision for large-scale visual search," *Signal Process., Image Commun.*, vol. 59, pp. 109–116, Nov. 2017.
[37] Z. Xia, X. Feng, J. Peng, and A. Hadid, "Unsupervised deep hashing for large-scale visual search," in *Proc. 6th Int. Conf. Image Process. Theory, Tools Appl. (IPTA)*, Dec. 2016, pp. 1–5.
[38] E. Yang, T. Liu, C. Deng, W. Liu, and D. Tao, "DistillHash: Unsupervised deep hashing by distilling data pairs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2946–2955.
[39] L. Yuan, T. Wang, X. Zhang, F. EH Tay, Z. Jie, W. Liu, and J. Feng, "Central similarity quantization for efficient image and video retrieval," 2019, *arXiv:1908.00347*. [Online]. Available: http://arxiv.org/abs/1908.00347
[40] L. Zhang, Y. Zhang, J. Tang, X. Gu, J. Li, and Q. Tian, "Topology preserving hashing for similarity search," in *Proc. 21st ACM Int. Conf. Multimedia*, 2013, pp. 123–132.
[41] Y. Zhang, Y. Feng, J. Shang, M. Zhou, and B. Qiang, "Attention-aware joint location constraint hashing for multi-label image retrieval," *IEEE Access*, vol. 8, pp. 3294–3307, 2020.
[42] H. Zhou, Y. Yuan, and C. Shi, "Object tracking using SIFT features and mean shift," *Comput. Vis. Image Understand.*, vol. 113, no. 3, pp. 345–352, Mar. 2009.
[43] 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. 2415–2421.
[44] H. Zhu and S. Gao, "Locality constrained deep supervised hashing for image retrieval," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 3567–3573.

**HYEONGU YUN** received the B.S. degree in electrical and computer engineering from Seoul National University, South Korea, in 2015, where he is currently pursuing the Ph.D. degree in electrical and computer engineering. His research interests include natural language processing and machine translation.

**YONGIL KIM** received the B.S. degree in electrical and computer engineering from Seoul National University, South Korea, in 2019, where he is currently pursuing the Ph.D. degree in electrical and computer engineering. His research interests include multimodal learning and image synthesis.

**TAEGWAN KANG** received the B.S. degree in electrical and computer engineering from Seoul National University, South Korea, in 2016, where he is currently pursuing the Ph.D. degree in electrical and computer engineering. His research interest includes the areas that have benefitted from adversarial attack in natural language processing.

**KYOMIN JUNG** (Member, IEEE) received the B.S. degree in mathematics from Seoul National University, Seoul, South Korea, in 2003, and the Ph.D. degree in mathematics from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2009. From 2009 to 2013, he was an Assistant Professor with the Department of Computer Science, KAIST. Since 2016, he has been an Assistant Professor and an Associate Professor with the Department of Electrical and Computer Engineering, Seoul National University (SNU), where he is currently an Adjunct Professor with the Department of Mathematical Sciences. His research interests include natural language processing, deep learning and applications, data analysis, and web services.