Contrastive Transformer Masked Image Hashing for Degraded Image Retrieval

Xiaobo Shen\textsuperscript{1}, Haoyu Cai\textsuperscript{1}, Xiwen Gong\textsuperscript{2} and Yuhui Zheng\textsuperscript{3,}\textasteriskcentered
\textsuperscript{1}Nanjing University of Science and Technology
\textsuperscript{2}University of Technology Sydney
\textsuperscript{3}Qinghai Normal University

njjust.shenxiaobo@gmail.com, hycai@njjust.edu.cn, gongxiwen@gmail.com, zhengyh@vip.126.com

Abstract

Hashing utilizes hash code as a compact image representation, offering excellent performance in large-scale image retrieval due to its computational and storage advantages. However, the prevalence of degraded images on social media platforms, resulting from imperfections in the image capture process, poses new challenges for conventional image retrieval methods. To address this issue, we propose Contrastive Transformer Masked Image Hashing (CTMIH), a novel deep unsupervised hashing method specifically designed for degraded image retrieval, which is challenging yet relatively less studied. CTMIH addresses the problem by training on transformed and masked images, aiming to learn transform-invariant hash code in an unsupervised manner to mitigate performance degradation caused by image deterioration. CTMIH utilizes Vision Transformer (ViT) architecture applied to image patches to capture distant semantic relevance. CTMIH introduces cross-view debiased contrastive loss to align hash tokens from augmented views of the same image and presents semantic mask reconstruction loss at the patch level to recover masked patch tokens. Extensive empirical studies conducted on three benchmark datasets demonstrate the superiority of the proposed CTMIH over the state-of-the-art in both degraded and normal image retrieval.

1 Introduction

Hashing [Wang \textit{et al.}, 2017] has been widely applied for efficient retrieval from large-scale image databases due to its superiority of low computation and storage costs. It aims to convert high-dimensional image features into low-dimensional compact hashing codes while preserving similarity structure among images. Many learning-based hashing methods [Weiss \textit{et al.}, 2008; Gong \textit{et al.}, 2012; Wang \textit{et al.}, 2017] have been proposed by employing machine learning on hash code generation, and are still being actively studied to support fast and accurate large-scale image retrieval.

There has been a recent research focus on deep hashing [Luo \textit{et al.}, 2023] that introduces deep learning [LeCun \textit{et al.}, 2015] into hashing, and learns image features and hash code in an end-to-end manner. With powerful learning capability of deep architectures, e.g., CNN [Krizhevsky \textit{et al.}, 2012], ViT [Dosovitskiy \textit{et al.}, 2021], deep hashing has shown its significant superiority over conventional shallow hashing [Weiss \textit{et al.}, 2008; Gong \textit{et al.}, 2012; Shen \textit{et al.}, 2015]. Supervised deep hashing [Cao \textit{et al.}, 2017; Li \textit{et al.}, 2017; Liu \textit{et al.}, 2019; Fan \textit{et al.}, 2020] employs pairwise semantics or class label information to supervise training, and has achieved promising performance in image retrieval. However, supervised deep hashing heavily requires manually annotated labels, which are expensive to collect. Unsupervised deep hashing without need of semantic labels is valuable yet challenging in real applications. Until now some efforts [Shen \textit{et al.}, 2020; Luo \textit{et al.}, 2021; Qiu \textit{et al.}, 2021; Ma \textit{et al.}, 2022; Yu \textit{et al.}, 2023] have been made towards unsupervised deep hashing.

Degraded images [Wang \textit{et al.}, 2020; Yang \textit{et al.}, 2022; Park \textit{et al.}, 2023] are commonly encountered in various sce-
narios due to imperfections in the image capture process. These imperfections can result in serious degradations such as flipping, color jitter, grayscale, and blur, as illustrated in Figure 1. In practical applications such as pedestrian monitoring using surveillance cameras, it is often necessary to use the degraded pedestrian images as queries to retrieve similar high-quality images from a large-scale face image database [Li et al., 2019]. The need for large-scale degraded image retrieval is evident, however this task is challenging and has received relatively less attention. Image degradation significantly affects semantic similarity of images, and may mislead training of conventional hashing, leading to reduced retrieval performance. Therefore, it is imperative to develop new deep hashing for degraded image retrieval, which remains relatively unexplored and presents substantial challenges.

To address this issue, we propose Contrastive Transformer Masked Image Hashing (CTMIH) that is specifically designed for degraded image retrieval. The key idea of CTMIH is to train hashing model on both transformed images and masked images to learn transform-invariant hash codes in an unsupervised manner, such that retrieval performance degradation caused by image degradation can be mitigated. The overview of the proposed CTMIH is illustrated in Figure 2. The main contributions of this work can be summarised as follows:

- We address the challenging task of degraded image retrieval, which has received relatively less attention. To tackle this, we propose a new deep unsupervised hashing method called Contrastive Transformer Masked Image Hashing (CTMIH).

CTMIH utilizes ViT to encode image patches and learns transform-invariant hash code by aligning augmented and masked hash tokens while recovering masked patch tokens.

- Extensive empirical evaluations conducted on three benchmark image datasets demonstrate the superior performance of the proposed method over the state-of-the-art in both degraded and normal image retrieval.

2 Related Work

Deep Hashing Due to the strong learning capability of advanced deep architectures, e.g., CNN [Krizhevsky et al., 2012], ViT [Dosovitskiy et al., 2021], deep hashing [Luo et al., 2023] has achieved promising performance in large-scale image retrieval. According to whether semantic supervision is used or not, deep hashing methods can be roughly divided into supervised deep hashing and unsupervised deep hashing. Supervised deep hashing [Cao et al., 2017; Li et al., 2017; Liu et al., 2019; Fan et al., 2020] typically outperforms unsupervised deep hashing by incorporating additional semantic labels. However, manual label annotation is time consuming and expensive, which is not often available in real applications.

Recently unsupervised deep hashing has been a hot research focus, as it overcomes the limitation of manual labeling. Some works [Dai et al., 2017; Shen et al., 2020] learn hash code by performing reconstruction tasks using some advanced deep architectures, e.g., variational autoencoders (VAE) [Kingma and Welling, 2013] and generative adversarial network (GAN) [Goodfellow et al., 2014]. For instance, Semantic Structure based unsupervised Deep Hashing (SSDH) [Yang et al., 2018] leverages two half Gaussian distributions to construct semantic structure, and further designs a pairwise similarity preservation loss. Bi-half
Net [Li and van Gemert, 2021] proposes to learn hash code by maximizing its bit entropy, and designs a parameter-free layer to force continuous image features to approximate the optimal half-half bit distribution. In addition, contrastive learning [Chen et al., 2020; He et al., 2020], as a powerful self-supervised method, has been introduced into deep unsupervised hashing, based on which some hashing methods have been proposed [Luo et al., 2021; Qiu et al., 2021; Ma et al., 2022]. Contrastive Quantization with Code Memory (McCoQ) [Wang et al., 2022] proposes to use contrastive loss to better capture discriminative visual semantics and further use quantization code memory to enhance contrastive learning with lower feature drift. While many deep hashing methods employ CNN as a backbone, there have been a few recent works [Qiu et al., 2021; Yu et al., 2022; Ng et al., 2023] that consider ViT as a backbone to encode images. Most existing deep unsupervised hashing methods have been developed for mainly conventional normal image retrieval, and may not perform well in degraded image retrieval, where image queries are degraded.

Degraded Image Analysis There have been a few attempts [Wang et al., 2020; Yang et al., 2022] on degraded image analysis, mainly on degraded image classification. Based on the observation that the distributions of corresponding patches in high- and low-quality images have uniform margins, feature de-drifting module (FDM) [Wang et al., 2020] is proposed to learn the mapping relationship between deep representations of high- and low-quality images, and is further leveraged as a deep degradation prior to degraded image classification. In addition, a self-feature distillation method with uncertainty modeling [Yang et al., 2022] is proposed for degraded image classification. It employs the high-quality features to distill the corresponding degraded ones and conduct uncertainty modeling, increasing importance of feature regions that are difficult to recover. However, to our knowledge, until now there have been few attempts of learning hash code for degraded image retrieval.

3 The Proposed Method

3.1 Problem Setup

Generally speaking, the goal of deep hashing for image retrieval is to map an ideal image $x$ to hash code $b$, which is used to support fast image retrieval in Hamming space. However, in real-world applications, due to various sources of degradation, e.g., noise, motion blur, and ColorJitter, we only observe the degraded image $\tilde{x}$ instead of the ideal one. This work develops deep hashing for a challenging yet less studied task, i.e., degraded image retrieval, where image queries are degraded, while images in database are normal.

In degraded image retrieval, given a degraded image query $\tilde{x}$, the aim of the proposed method is to learn a deep hash function $H : \tilde{x} \rightarrow b \in \{-1, 1\}^L$, where $b$ is the hash code, $L$ is code length. The learned hash code is then employed to achieve fast and accurate retrieval of the degraded image query from a normal image database. As the degraded images have different inherent appearances from normal images, it can result in obvious retrieval performance degradation using the conventional hash function. To address this issue, the key of the proposed method is to learn the transform-invariant hash function that is robust to image degradation. As such, the degraded image has similar hash code to its normal image, making it possible to quickly and accurately retrieve images that are visually or semantically similar to the degraded image query.

3.2 Formulation

Architecture The most common and natural solution for mitigating retrieval performance degradation caused by image degradation is to train a deep hashing model with augmented images having various image transformations. For a given image $X$, the two random augmentations are applied, yielding two distorted views: $U = T(X, \delta_u)$ and $V = T(X, \delta_v)$, where $T$ denotes the transformation function that performs image degradation, hyper-parameter $\delta_u$ and $\delta_v$ control the degrees of the two transformation. In this work, the transformations include random crop, resize, flip, color jitter, and blur, and the transformation procedure is illustrated in Algorithm 1. It is clear that large $\delta$ leads to heavy image degradation. In addition, motivated by the success of masked image modeling, the blockwise masking is applied to two views $U$ and $V$ to obtain masked views $\hat{U}$ and $\hat{V}$ respectively. Specifically, for instance, given $U$, we first divide it into $K$ non-overlapping patches $u_k$, where $k = 1, \ldots, K$, and then, with a masking ratio $r$, perform masking on a random subset of patches, which are replaced by $e$.

As illustrated in Figure 2, Siamese network structure with two branches sharing network weights is employed in CT-MIH, and the patches of augmented views and masked views are fed into the two branches. Existing deep hashing methods mainly employ CNN as the backbone that performs convolution operations on a small neighborhood of an image, and struggles to relate concepts spatially apart. Vision Transformer (ViT) [Dosovitskiy et al., 2021] can effectively capture distant semantic relevance in an image by applying self-attention to a series of patches in an image. Inspired by the powerful capability of ViT on image modeling, we propose to employ ViT as an encoder in each branch, which receives image patches as sequential data and generates their latent representations. Following ViT encoder, a hash layer consists of a fully-connected layer and the sign function. For instance, given augmented view $U$ and its masked view $\hat{U}$, hash layer outputs their hash tokens $h^{u}$ and $h^{\hat{u}}$, and patch tokens $p^{u}_k$ and $p^{\hat{u}}_k$, where $k = 1, \ldots, K$, and finally binarizes $h^{u}$ to obtain hash code $b^{u} = \text{sign}(h^{u})$.

**Algorithm 1 Image Transformation $T$**

**Input:** Image $X$, hyper-parameter $\delta$;  
**Output:** Transformed Image.

1. Crop $X$ with size of $256 \times 256$ randomly;
2. Resize $X$ to size of $224 \times 224$;
3. Flip $X$ horizontally with a probability of $0.5\delta$;
4. Add ColorJitter to $X$ with a probability of $0.8\delta$;
5. Convert $X$ to grayscale image with a probability of $0.4\delta$;
6. Apply Gaussian blur to $X$ with a probability of $0.5\delta$. 

Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24)
Cross-view Debiased Contrastive Learning  For conventional hashing methods, a degraded image tends to produce different hash code compared to its original counterpart due to changes in appearance, leading to unsatisfactory performance in degraded image retrieval. To mitigate this issue, a possible solution is to leverage contrastive learning (CL) [Chen et al., 2020; He et al., 2020] that aims to distinguish between semantically similar and dissimilar pairs of samples. Based on a triplet set that consists of anchor, positive, and negative samples, CL encourages the anchor and positive sample to be pulled closer together while pushing away the anchor from the negative samples. This work considers the benefit of CL that can well align the hash tokens of the two views augmented from one image, and push away the hash tokens of different images.

Specifically, given the i-th image X_i, the hash tokens of its two augmented views are denoted as h^u_i and h^v_i, and its two masked views are denoted as h^u_i, h^v_i. Taking h^u_i as anchor, we regard another masked view h^v_i as positive sample, and (h^u_j)_{j=1,j\neq i} as negative samples, where n is batch size. In this work, we contrast augmented and masked views with the purpose of distilling knowledge of unmasked views to help learn the hash token of masked views. Our idea is to enable the positive pairs to be close and the negative pairs to be apart. Treating h^u_i as anchor, we minimize the following conventional CL loss:

$$-\log \frac{o(h^u_i, h^v_i)}{o(h^u_i, h^u_j) + \sum_{j=1, j\neq i} o(h^u_i, h^v_j)}$$ \hspace{1cm} (1)$$

where function $o(h^u_i, h^v_i) = \exp(h^u_i \cdot h^v_i / \tau)$ measures the similarity between two embeddings, $\tau$ is a temperature coefficient that controls dynamic range of product. The $\ell_2$ normalization is applied on input embeddings before computing the inner product, such that the inner product is equivalent to cosine similarity.

As can be seen in (1), in conventional CL, negative pairs are randomly sampled from the whole set as true labels are unavailable. However, some of these negative samples may actually belong to the same class as anchor, referred as sampling bias, and this bias has been shown to degrade performance. To mitigate this gap, debiased CL is employed to correct sampling bias of negative samples, and its loss is defined as follows:

$$J_C(h^u_i, h^v_i) = -\log \frac{o(h^u_i, h^v_i)}{o(h^u_i, h^u_j) + \sum_{j=1, j\neq i} o(h^u_i, h^v_j)} + n\varphi(h^u_i, (h^v_j)_{j=1}^{n-1})$$ \hspace{1cm} (2)$$

where the negative similarity term can be modified as:

$$\varphi(h^u_i, (h^v_j)_{j=1}^{n-1}) = \frac{1}{\varphi^- + \varphi^+} \left( \frac{1}{n} \sum_{j=1}^{n} o(h^u_i, h^v_j) - g^+ o(h^u_i, h^v_i) \right)$$ \hspace{1cm} (3)$$

where $\varphi^-$ and $\varphi^+$ represent the class probability that two images belong to same and different classes respectively, and we have $\varphi^- + \varphi^+ = 1$. Similarly, we can further define $J_C(h^u_i, h^v_i)$.

To this end, by considering the $n$ images, we have the following debiased CL loss:

$$J_C = \frac{1}{n} \sum_{i=1}^{n} \left( J_C(h^u_i, h^v_i) + J_C(h^v_i, h^u_i) \right)$$ \hspace{1cm} (4)$$

Semantic Mask Reconstruction Learning  Masked image patch reconstruction is a popular self-supervised pretext task with the idea of auto-encoding [Zhou et al., 2021], and has been previously achieved by predicting raw pixels. This work leverages the benefits of masked image modeling to explore internal local structures in an image and better train ViT. Specifically, for instance, given one augmented view $U_i$ and the corresponding masked view $\hat{U}_i$ of the i-th image, their patch token sequences are denoted as $\{p^u_{ik}\}_{k=1}^{K}$ and $\{p^\text{φ}_i\}_{k=1}^{K}$. In this work, we employ the idea of self-distillation, and instead of recovering raw pixels, propose to semantically recover the masked patch token using original patch token, i.e., enable the masked and its original tokens to be close. Specifically, given the i-th image, we recover its k-th image patch by minimizing the following cross-entropy (CE) loss:

$$J_R(p^u_{ik}, p^\text{φ}_i) = -m_{ik} p^\text{φ}_i \log p^u_{ik}$$ \hspace{1cm} (5)$$

where $m_{ik}$ is set to 1 if the k-th patch of the i-th image is masked, and set to 0 otherwise. It simplifies masked recovering by turning it into a classification problem that is optimized by CE loss, and preserves more semantic information of patch token.

By considering all the image patches of all the $n$ images, we have the following semantic mask reconstruction loss:

$$J_R = \sum_{i=1}^{n} \sum_{k=1}^{K} J_R(p^u_{ik}, p^\text{φ}_i) + J_R(p^\text{φ}_i, p^u_{ik})$$ \hspace{1cm} (6)$$

3.3 Training and Inference

Training  To obtain high-quality hash code, quantization loss is introduced to reduce quantization error. Motivated by the observation that hashing aims to predict the sign of each bit, this problem can be naturally regarded as the binary classification. Specifically, we employ a pre-defined Gaussian distribution estimator $g(h) = \exp(-\frac{(h-\mu)^2}{2\sigma^2})$ to evaluate binary likelihood of each hash bit, where $\mu$ and $\sigma$ denote mean and standard deviation respectively. We then define $G(\cdot) = \{g^+, g^-\}$, where with the same $\sigma$, $g^+$ and $g^-$ are defined with $\mu = 1$ and $\mu = -1$ respectively. The quantization loss is calculated as binary cross-entropy classification loss (BCE). Taking $h^u_i$ for example, its quantization loss is defined as follows:

$$J_Q(h^u_i) = \frac{1}{L} \sum_{l=1}^{L} \left( Q(y^+_l, q^+_l) + Q(y^-_l, q^-_l) \right)$$ \hspace{1cm} (7)$$

where BCE loss is defined as $Q(y, g) = -y \log g + (1 - y) \log (1 - g)$, $q^+_l = g^+(h^u_i)$ and $q^-_l = g^-(h^u_i)$ denote the two estimated likelihoods of l-th hash bit of $h^u_i$, $y^+_l = \frac{1}{2}(\text{sign}(h^u_i) + 1)$ and $y^-_l = 1 - y^+_l$ denote the likelihood labels, $L$ is the code length. In this way, the quantization error
is reduced by minimizing the above binary classification loss. Considering the hash tokens of all the \( n \) images, we have the following quantization loss:

\[
J_Q = \sum_{i=1}^{n} J_Q(h_i^x) + J_Q(h_i^y) \tag{8}
\]

To this end, by summarizing the three losses, i.e., \( J_C \), \( J_R \), \( J_Q \), we have the following objective function of the proposed method:

\[
J = J_C + \alpha J_R + \beta J_Q \tag{9}
\]

where \( \alpha \) and \( \beta \) are two non-negative parameters to balance each term.

Inference Once the proposed CTMIH is trained, and given an image query, we first divide it into several patches and further feed it into the ViT encoder and hash layer without masking to generate its hash code. For degraded image retrieval, the image query is required to be degraded. We perform transformations on a normal image to obtain a degraded image query, which is detailed in the experiment setup.

4 Experiments

4.1 Experimental Setup

Datasets The experiments are conducted on three benchmark image datasets, which are detailed as follows:

MSCOCO [Lin et al., 2014] is a large-scale image dataset for object detection, segmentation, and captioning. Following previous works, a subset of 122,218 images from 80 categories is used, where the 5,000 images are randomly selected as the query set and the remaining images are used as the database. The 10,000 images are randomly selected from the database for training.

NUS-WIDE [Chua et al., 2009] is a multi-label dataset consisting of 269,648 images from 81 categories. A subset with images from the 21 most frequent categories is used. The 100 images are randomly sampled from each category as the query set and the remaining images are used as the database. The 500 images for each category are randomly sampled from the database for training.

ImageNet [Russakovsky et al., 2014] is a single-label image dataset, where each image is labeled by one of 1,000 categories. A subset with 143,495 images in 100 categories is used, where 100 images from each category are randomly sampled for training, 5,000 images are sampled as the query set, and the remaining images are used as the database.

Baselines We compare the proposed method with various state-of-the-art unsupervised hashing baselines, including two shallow hashing methods, i.e., SH [Weiss et al., 2008] and ITQ [Gong et al., 2012], five CNN based deep hashing methods, i.e., SSDH [Yang et al., 2018], Greedy-Hash [Su et al., 2018], Bi-half Net [Li and vanGemert, 2021], MeCoQ [Wang et al., 2022], OH [Yu et al., 2023], three ViT based deep hashing methods, i.e., CIBHash [Qu et al., 2021], WCH [Yu et al., 2022], SDC [Ng et al., 2023].

Experimental Setting For all the methods, the images are resized to 224 \( \times \) 224 \( \times \) 3. For shallow hashing methods, the 4,096-dimensional feature extracted by the VGG-F model pre-trained on ImageNet is used for training. For deep hashing methods, the raw image is directly used as for training. For the proposed method, we apply Algorithm 1 on each image in the training set to generate two augmented views, where \( \delta_a \) and \( \delta_t \) are set to 0.5 and 1 respectively. To generate the degraded image query, we apply Algorithm 1 on each image in the query set by setting \( \delta \) to 0.5 by default. The standard ViT-Base is used as the backbone, and the size and number of patches are set to 16 and 196 respectively. The masking ratio \( r \) is set to 0.3, class probability \( \varphi \) is set to 0.05, and temperature \( \tau \) is set to 0.5. The two hyper-parameters \( \alpha \) and \( \beta \) are set to 0.1 and 0.1 respectively. The batch size is set to 32, the number of epochs is set to 100, and the learning rates of ViT and hash layer are set to \( 10^{-3} \) and \( 10^{-3} \) respectively. The proposed method is trained using Adam optimizer.

Evaluation Metrics Following [Zhang et al., 2017], we consider the widely-used mean Average Precision (mAP) and Precision curve as evaluation metrics, and \( K \) is set to 5000 for NUS-WIDE and MSCOCO, and 1000 for ImageNet.

4.2 Comparisons with State-of-the-art

Results on Degraded Image Retrieval This section evaluates the performance of degraded image retrieval. The mAPs of the proposed CTMIH and ten state-of-the-art hashing baselines on three benchmark datasets are reported in Table 1. In addition, precision curves of all the methods with respect to
32 bits are shown in Figure 3. From Table 1 and Figure 3, we can clearly observe that 1) the proposed CTMIH outperforms all the baselines in all the 10 cases. Specifically, it outperforms the baselines averagely by 3.3%, 1.1%, 7.4% on MS COCO, NUS-WIDE, ImageNet respectively. In addition, the precision curves of CTMIH are generally above those of the baselines. 2) among all the baselines, CIBHash, WCH, SDC with VIT backbone outperform the other deep hashing baselines with VGG backbone by a margin, followed by shallow hashing baselines. The empirical results clearly demonstrate the superiority of the proposed CTMIH for degraded image retrieval.

**Results on Normal Image Retrieval** This section evaluates the performance of conventional normal image retrieval, where the image queries are normal. The mAPs of all the hashing methods with respect to 16 bits are reported in Table 2. From this table, we see that the proposed CTMIH outperforms all the baselines, indicating that the proposed method also works well on conventional normal image retrieval. In addition, compared to normal image retrieval, the mAP drops of the proposed method on degraded image retrieval are lower than those of the baselines. This suggests that the proposed CTMIH is superior in mitigating the performance degradation caused by image degradation.

### 4.3 Further Analysis

#### Evaluation on Varying Degrees of Image Degradation

The section evaluates the sensitivity of the deep hashing methods with respect to varying degrees of image degradation. We vary the parameter $\delta$ from the range of $[0, 0.2, 0.4, 0.6, 0.8, 1.0]$ to generate image queries with varying degrees of degradation, where the sample image query is shown in Figure 4. Figure 4 reports the mAPs of all the deep hashing methods with varying degrees of degradation on ImageNet. From this figure, we can clearly observe that mAP of the proposed CTMIH is not relatively insensitive to the change of $\delta$, compared to the deep hashing baselines. Specifically, as the degree of image degradation increases, the mAP of the proposed CTMIH decreases by 1.3%, while the mAPs of the baselines have shown a decrease ranging from 5.2% to 20.1%. The above results clearly demonstrate that the proposed CTMIH is robust to image degradation, and performs well on degraded image retrieval.

**Ablation Study** This section empirically evaluates the effectiveness of each loss in the proposed CTMIH. We compare the proposed method with a baseline that is trained with conventional contrastive loss without masking strategy, and its effectiveness of each loss in the proposed CTMIH. We compare the proposed method with a baseline that is trained with conventional contrastive loss without masking strategy, and its effectiveness of each loss in the proposed CTMIH.

---

**Table 2:** mAPs of all the hashing methods for normal image retrieval on the three benchmark datasets. The mAP drops on degraded image retrieval compared to normal image retrieval are also reported.

<table>
<thead>
<tr>
<th>Method</th>
<th>MS COCO</th>
<th>NUS-WIDE</th>
<th>ImageNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH</td>
<td>0.585 (\pm) 0.061</td>
<td>0.638 (\pm) 0.110</td>
<td>0.338 (\pm) 0.124</td>
</tr>
<tr>
<td>ITQ</td>
<td>0.658 (\pm) 0.060</td>
<td>0.718 (\pm) 0.102</td>
<td>0.439 (\pm) 0.112</td>
</tr>
<tr>
<td>SSDH</td>
<td>0.556 (\pm) 0.016</td>
<td>0.698 (\pm) 0.023</td>
<td>0.229 (\pm) 0.053</td>
</tr>
<tr>
<td>GreedyHash</td>
<td>0.586 (\pm) 0.037</td>
<td>0.675 (\pm) 0.038</td>
<td>0.231 (\pm) 0.084</td>
</tr>
<tr>
<td>Bi-half Net</td>
<td>0.645 (\pm) 0.007</td>
<td>0.750 (\pm) 0.014</td>
<td>0.540 (\pm) 0.101</td>
</tr>
<tr>
<td>MeCoQ</td>
<td>0.755 (\pm) 0.063</td>
<td>0.770 (\pm) 0.026</td>
<td>0.705 (\pm) 0.051</td>
</tr>
<tr>
<td>OH</td>
<td>0.748 (\pm) 0.092</td>
<td>0.796 (\pm) 0.085</td>
<td>0.672 (\pm) 0.018</td>
</tr>
<tr>
<td>CIBHash</td>
<td>0.793 (\pm) 0.026</td>
<td>0.790 (\pm) 0.009</td>
<td>0.783 (\pm) 0.051</td>
</tr>
<tr>
<td>WCH</td>
<td>0.760 (\pm) 0.019</td>
<td>0.788 (\pm) 0.007</td>
<td>0.761 (\pm) 0.028</td>
</tr>
<tr>
<td>SDC</td>
<td>0.810 (\pm) 0.045</td>
<td>0.787 (\pm) 0.023</td>
<td>0.764 (\pm) 0.030</td>
</tr>
<tr>
<td>CTMIH</td>
<td>0.818 (\pm) 0.009</td>
<td>0.799 (\pm) 0.004</td>
<td>0.832 (\pm) 0.012</td>
</tr>
</tbody>
</table>

**Table 3:** The mAPs of the proposed method and four variants on three datasets.

<table>
<thead>
<tr>
<th>Method</th>
<th>MS COCO</th>
<th>NUS-WIDE</th>
<th>ImageNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive CL</td>
<td>0.719</td>
<td>0.698</td>
<td>0.674</td>
</tr>
<tr>
<td>w/o $J_Q$</td>
<td>0.765</td>
<td>0.731</td>
<td>0.757</td>
</tr>
<tr>
<td>w/o $J_R$</td>
<td>0.792</td>
<td>0.779</td>
<td>0.806</td>
</tr>
<tr>
<td>w/o $J_C$</td>
<td>0.428</td>
<td>0.535</td>
<td>0.413</td>
</tr>
<tr>
<td>CTMIH</td>
<td>0.809</td>
<td>0.795</td>
<td>0.820</td>
</tr>
</tbody>
</table>
of these methods with respect to 16 bits on the three datasets are reported in Table 3. From Table 3, we can clearly observe that removing $J_Q$, $J_R$, $J_C$ results in an average decrease of 5.7%, 1.6%, 34.9% in mAP respectively on the three datasets, where the debiased contrastive loss has the greatest impact on performance. In addition, CTMIH obviously outperforms naive CL baseline by 11.10% on average.

**Embedding Visualization** This section qualitatively compares different hashing methods by visualizing learned hash codes of degraded images. We conduct experiment on ImageNet, randomly selecting 10,000 samples that belong to 10 classes, and setting code length to 32. The hash codes learned by ten methods are visualized into a 2-dimensional space with t-SNE [van der Maaten and Hinton, 2008], as illustrated in Figure 5. From Figure 5, we see that the visualization results are generally consistent with the quantitative empirical results.

**Attention Map Visualization** This section qualitatively evaluates the effectiveness of the proposed CTMIH by visualizing its attention maps. Figure 6 illustrates the attention maps generated by CTMIH for 5 randomly selected degraded images from ImageNet. From the figure, we can observe that the generated attention maps focus on the key components of images through masking reconstruction task. It indicates that the proposed CTMIH can effectively identify and prioritize important regions in the image.

**Case Study** This section presents a case study comparing the performance of the proposed CTMIH with three baselines, namely CIBHash, WCH, and SDC, for both degraded and normal image retrieval tasks. Figure 7 illustrates the top-10 retrieved images of one randomly selected normal image and its degraded image queries from ImageNet. A retrieved sample is marked in green if its label matches that of the query, and it is marked in red otherwise. From the results shown, we can observe that CTMIH outperforms the three baselines in terms of retrieving correct images in both tasks. These findings further confirm the effectiveness of the proposed CTMIH for both degraded and normal image retrieval tasks.

5 Conclusions

This work presents a preliminary attempt to learn hash code from degraded images, and proposes Contrastive Transformer Masked Image Hashing (CTMIH) for this challenging yet less studied degraded image retrieval. The proposed CTMIH aims to learn transform-invariant hash code through unsupervised training on transformed and masked images, mitigating performance degradation caused by image degradation. CTMIH aligns hash tokens of augmented views from the same image by performing constrastive learning, and recovers masked patch tokens with the idea of masked image prediction. The extensive empirical studies demonstrate the proposed CTMIH performs effectively on both degraded and normal image retrieval.
Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 62176126, U20B2065, U22B2056, 62272468, the Natural Science Foundation of Jiangsu Province, China under Grant No. BK20230095, BK20211539.

References


