

引用格式: 吴楠楠, 杨宵晗, 刘文皓, 等. 非对称深度在线哈希 [J]. 电子科技大学学报, 2025, 54(1): 103-115.

WU N N, YANG X H, LIU W H, et al. Asymmetric deep online Hashing[J]. Journal of University of Electronic Science and Technology of China, 2025, 54(1): 103-115.



非对称深度在线哈希

吴楠楠¹, 杨宵晗¹, 刘文皓¹, 常心怡¹, 郭呈银¹, 王 振^{1,2*}

(1. 山东理工大学 计算机科学与技术学院, 淄博 255000; 2. 吉林大学 符号计算与知识工程教育部重点实验室, 长春 130012)

摘要: 海量图像以流数据的形式实时涌入网络, 使得在线图像检索需求越来越迫切。为了保证在线图像检索性能, 研究人员利用在线哈希算法实时更新哈希函数, 并重新学习新、旧数据集的哈希码。然而, 随着旧数据集的日益积累, 在线更新旧数据集的哈希码会严重影响在线检索效率。为此, 提出非对称深度在线哈希 (asymmetric deep online Hashing, ADOH), 以非对称的方式深度学习在线哈希网络, 并且仅生成新数据集的哈希码, 无须更新旧数据集的哈希码, 能够有效地提升在线检索效率。ADOH 算法通过最小化哈希码内积与相似度矩阵之间的差异, 保持样本对之间的语义相似性关系。另外, ADOH 算法建立分类损失项和标签嵌入模块学习样本的语义信息, 使生成的哈希码更具备语义鉴别性。在 3 个广泛使用的数据集 cifar-10、mnist 和 Places205 上设置在线近邻检索对比实验, 结果表明 ADOH 算法的在线近邻检索性能优于目前 8 种较先进的在线哈希算法。

关键词: 图像检索; 在线哈希; 深度学习; 非对称学习

中图分类号: TP391.4

文献标志码: A

DOI: 10.12178/1001-0548.2023170

Asymmetric deep online Hashing

WU Nannan¹, YANG Xiaohan¹, LIU Wenhao¹, CHANG Xinyi¹, GUO Chengyin¹, and WANG Zhen^{1,2*}

(1. School of Computer Science and Technology, Shandong University of Technology, Zibo 255000, China;

2. Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China)

Abstract: Massive images are flooding into the internet in real-time in the form of streaming data, which makes the need of online image retrieval more and more urgent. To guarantee the online image retrieval performance, online Hashing algorithms are utilized to re-learn the Hash functions and re-generate the Hash codes of the new and old samples in real time. As time went by, the amount of old dataset is very large, and the time complexity of re-generating Hash codes become unacceptable. To avoid the above problems, we propose a novel asymmetric deep online Hashing (ADOH) which trains a deep online Hashing network in an asymmetric manner. To improve the online retrieval efficiency, ADOH only generates the Hash code of the new samples and do not update the old samples' Hash codes. During the training process, ADOH minimizes the difference between the Hash codes' inner product and the similarity matrix, which can preserve the pair-wise semantic similarity relationship. Moreover, ADOH establishes the classification loss and the label embedding model to learn the samples' semantic information, which makes the generated Hash codes more semantically discriminative. We conduct the approximate nearest neighbor retrieval comparative experiments on three widely used datasets including the cifar-10 dataset, the mnist dataset and the Places205 dataset. The results show that the online approximate nearest neighbor retrieval performance of ADOH outperforms the other 8 existing online Hashing methods.

Key words: image retrieval; online Hashing; deep learning; asymmetric learning

随着信息技术的不断发展, 智能通信设备的便携化, 海量多媒体数据, 如音频、视频、图像等, 以流数据的形式不断地涌入网络。面对实时增长的图像数据, 如何快速响应用户的检索请求, 是当前

在线图像检索任务面临的一大挑战^[1]。基于哈希的近似最近邻检索技术^[2-6]将高维浮点数据映射为紧凑二进制编码, 并在汉明空间内保持样本之间的原始相似性关系, 从而能根据样本之间的汉明距离关

收稿日期: 2023-06-14

基金项目: 国家自然科学基金 (61841602); 山东省自然科学基金 (ZR2020QF069, ZR2021MF017)

作者简介: 吴楠楠, 主要从事计算机视觉方面的研究。

*通信作者 E-mail: wzh@sdut.edu.cn

系快速检索近邻样本。凭借计算复杂度低、占用存储少的优点,基于哈希的近似最近邻检索技术已被广泛用于大规模图像检索任务。

对于在线图像检索实际网络场景而言,图像数据并不是一次性生成的,而是以流数据的形式源源不断地涌入。随着时间的推移,不断出现新数据,使得数据库实时发生变化,无法得到稳定的数据库样本集。传统哈希算法离线学习哈希函数^[1],若有新数据涌入,需要根据所有新、旧数据重新训练哈希函数,时间复杂度较高,无法适用于流数据的在线近邻检索任务。相对而言,在线哈希算法仅根据新数据流更新哈希函数,能够有效降低在线近邻检索复杂度。

近年来,为了获得较优的在线近邻检索性能,越来越多的在线哈希算法被相继提出。在线监督哈希^[7](online supervised Hashing, OSH)、基于哈达玛码本在线哈希^[8](hadamard codebook based online Hashing, HCOH)和基于哈达玛矩阵在线哈希^[9](hadamard matrix guided online Hashing, HMOH)提出两步学习策略,先为数据分配合适的纠错输出编码^[10](error correcting output codes, ECOC),再学习哈希函数,从而增强哈希函数对未知标签数据的适应能力。平衡相似性在线离散哈希^[11](balanced similarity for online discrete Hashing, BSODH)、基于可扩展监督在线哈希^[12](scalable supervised online Hashing for image retrieval, SSOH)和类智能快速在线哈希^[13](fast class-wise updating for online Hashing, FCOH)通过最小化哈希码内积与相似度矩阵之间的距离,建立新、旧数据之间的联系,使新、旧数据对在语义空间和原始特征空间具有相同的相似性关系。综上,在线哈希算法在适应未知标签数据、相似性保持等方面取得了较大进展。然而,面向实际网络图像在线近邻检索任务,现有在线哈希算法仍存在以下问题:1)在线哈希算法根据新数据集更新哈希函数后,需要同时更新旧数据集的哈希码,更新复杂度较高;2)仅关注保持新、旧数据对之间的相似性关系,无法捕捉单一数据点的语义信息;3)浮点特征生成和哈希函数训练过程相互独立,二者之间的适应性较差。

为了解决上述问题,本文提出非对称深度在线哈希(asymmetric deep online Hashing, ADOH)。

1 相关工作

哈希算法能够快速响应大规模图像检索任务,引起越来越多研究人员的关注。根据哈希函数的训练阶段,可将现有哈希算法大致分为离线深度哈希

算法和在线哈希算法。

离线深度哈希算法通常建立端到端的学习框架,同时生成浮点特征和紧凑二值编码,并可进一步分为对称深度哈希算法^[14-16]和非对称深度哈希算法^[17-19]。深度成对监督哈希^[14](deep pairwise supervised Hashing, DPSH)提出浮点特征和哈希码的统一学习框架,避免浮点特征与哈希码相互独立的问题。深度监督离散哈希^[15](deep discrete supervised Hashing, DDSH)将成对标签作为监督信息,离散监督训练深度哈希网络。深度学习监督哈希^[16](deep learning based supervised Hashing, DLSH)集成深度网络模块和哈希码学习模块,直接生成哈希码,可有效减少松弛损失。对称深度哈希算法利用全部数据集训练深度哈希网络,随着数据集的不断积累,将导致深度哈希网络的训练时间难以接受。为了解决上述问题,非对称深度哈希算法^[17]提出只根据查询样本更新深度哈希网络,有效降低训练时长。深度语义非对称哈希^[18](deep semantic asymmetric Hashing, DSAH)提出充分挖掘标签的语义信息,更准确地表示样本之间的相似性关系,从而生成更具鉴别性的二进制编码。基于非对称哈希学习的遥感图像检索算法^[19](asymmetric Hash code learning for remote sensing image retrieval, AHCL)在非对称学习框架中引入标签预测,进一步提升深度浮点特征的性能。非对称深度哈希算法以非对称的方式处理查询样本和数据库样本,提高了时间效率。

在线哈希算法根据新数据流实时学习哈希网络并更新相关哈希码,适用于有新数据流实时涌入的在线检索任务。根据训练过程中是否引入监督信息,可将在线哈希算法大致分为无监督在线哈希算法和监督在线哈希算法。在线草图哈希^[20](sketching Hashing, SketchHash)和快速在线草图哈希^[21](faster online sketching Hashing, FROSH)均为无监督在线哈希算法,其利用数据草图矩阵压缩存储数据,提高存储空间利用率。监督在线哈希算法利用语义标签作为监督信息,挖掘样本之间的语义关系,提高检索精度。在线核哈希^[22](online kernel Hashing, OKH)和自适应哈希^[23](adaptive Hashing, AdaptHash)以成对数据标签监督训练哈希函数,能够保留新、旧数据集的主要信息。互信息在线哈希^[24](online Hashing with mutual information, MIHash)利用邻居与非邻居之间的互信息,可减少汉明空间内邻居邻域的模糊性。OSH^[7]通过ECOC^[10]监督训练哈希函数,使新增类别的数据适应于在线哈希模型。基于HMOH^[9]和HCOH^[8]将

哈达玛矩阵作为纠错输出码, 并为数据集离线分配哈达玛编码, 能够提高在线学习的鲁棒性。BSODH^[11] 通过非对称图构建新、旧数据集之间的相似性关系, 并离散优化目标函数, 从而有效提升优化效率。FCOH^[13] 提出以类别的方式快速学习哈希函数, 能够解决训练效率低和自适应差的问题。高效更新哈希码在线哈希^[25] (online Hashing with efficient updating of binary codes, OHWEU) 通过投影函数在线生成流数据的哈希码, 其能固定哈希函数, 提高哈希码的生成效率。

在线哈希算法注重效率, 若将神经网络应用于在线哈希算法, 需要特别关注提高深度哈希网络的训练效率。受非对称深度哈希算法的启发, 本文提出一种高效的深度在线哈希算法 ADOH, 设计了非对称学习方式, 只根据采样数据集训练深度哈希网络, 并直接通过最小化目标损失函数计算新数据的哈希码, 避免了旧数据集哈希码的重复更新问题, 从而有效提升了深度在线哈希算法的检索效率。

2 非对称深度在线哈希

2.1 算法框架和符号表示

ADOH 算法主要设计了 3 个流程: 1) 根据采样数据集训练非对称深度在线哈希网络; 2) 利用标签嵌入模块学习新、旧数据集样本的语义信息; 3) 通过最小化目标损失函数, 直接计算新数据集对应的哈希码。

ADOH 算法流程如图 1 所示, 图像数据集被划分为旧、新两种数据集, 并分别随机取样得到采样数据集。首先, 对于旧数据集以及采样数据集, ADOH 算法建立了传统深度哈希网络, 分类损失项利用 AlexNet^[26] 网络生成浮点特征, 并在其之后建立全连接层作为哈希函数层, 用于学习数据集的哈希码。其次, 对于采样数据集, ADOH 算法建立分类损失项, 在深度哈希网络层之后建立了一层 softmax 激活函数全连接层, 作为语义网络层。其可生成样本的语义标签概率分布, 并通过最小化交叉熵损失, 确保语义一致性。进一步, ADOH 算法建立了汉明语义一致性模块, 要求最小化新、旧数据集与采样数据集的哈希码内积与语义标签相似度矩阵之间的距离, 使数据集在汉明空间和语义空间内具有相同的相似性关系。再者, 对于新、旧数据集, ADOH 算法通过标签嵌入模块, 将新、旧数据集标签嵌入待学习的哈希码中, 确保同类数据的哈希码趋于一致。最后, 对于新数据集, ADOH 算法直接通过最小化目标函数, 生成新数

据集的哈希码。

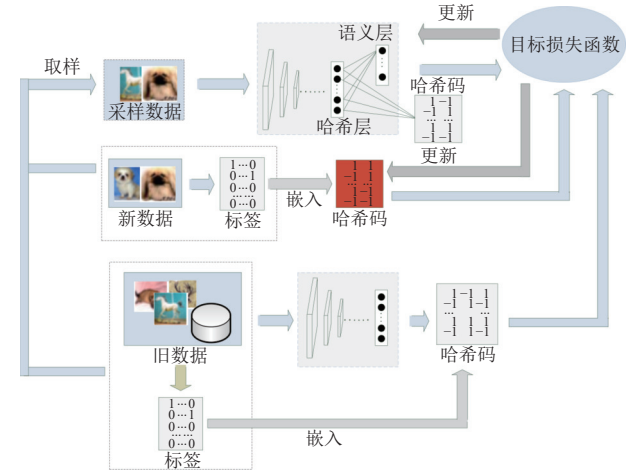


图 1 非对称深度在线哈希框架图

为了方便之后的符号描述, 在表 1 中整理了本文涉及的符号表示及含义。

表 1 符号表示及含义

| 符号 | 含义 |
|---------|--|
| X | 图像数据集 $X = \{X_s^t, X_e^t\}^n$ 中包含 n 个样本 |
| L | $L = \{L_s^t, L_e^t\} \in \{1, 0\}^{c \times n}$ 表示数据集 X 对应的标签矩阵 |
| X_s^t | $X_s^t = \{x_{s_1}^t, x_{s_2}^t, \dots, x_{s_{m_t}}^t\}^{m_t}$ 表示第 t 批次新数据集 |
| L_s^t | $L_s^t = \{l_{s_1}^t, l_{s_2}^t, \dots, l_{s_{m_t}}^t\} \in \{1, 0\}^{c \times m_t}$ 表示第 t 批次 X_s^t 对应的标签矩阵 |
| B_s^t | $B_s^t = \{b_{s_1}^t, b_{s_2}^t, \dots, b_{s_{m_t}}^t\} \in \{-1, 1\}^{k \times m_t}$ 表示第 t 批次 X_s^t 待学习的哈希码 |
| X_e^t | $X_e^t = \{x_{e_1}^t, x_{e_2}^t, \dots, x_{e_{(n-m_t)}}^t\}^{n-m_t}$ 表示第 t 批次旧数据集 |
| L_e^t | $L_e^t = \{l_{e_1}^t, l_{e_2}^t, \dots, l_{e_{(n-m_t)}}^t\} \in \{1, 0\}^{c \times (n-m_t)}$ 表示第 t 批次 X_e^t 对应的标签矩阵 |
| B_e^t | $B_e^t = \{b_{e_1}^t, b_{e_2}^t, \dots, b_{e_{(n-m_t)}}^t\} \in \{-1, 1\}^{k \times (n-m_t)}$ 表示第 t 批次 X_e^t 对应的哈希码 |
| X_q^t | $X_q^t = \{x_{q_1}^t, x_{q_2}^t, \dots, x_{q_d}^t\}^d \subset \{X_s^t, X_e^t\}^n$ 表示新、旧数据集中随机选取样本得到采样数据集 |
| L_q^t | $L_q^t = \{l_{q_1}^t, l_{q_2}^t, \dots, l_{q_d}^t\} \in \{1, 0\}^{c \times d} \subset \{L_s^t, L_e^t\}^n$ 表示 X_q^t 对应的标签矩阵 |
| B_q^t | $B_q^t = \{b_{q_1}^t, b_{q_2}^t, \dots, b_{q_d}^t\} \in \{-1, 1\}^{k \times d}$ 表示 X_q^t 对应的哈希码 |
| n | 数据集样本数目 |
| m_t | 新数据集样本数目 |
| d | 采样数据集样本数目 |
| k | 哈希码的长度 |
| c | 样本类别数目 |

2.2 非对称深度在线哈希网络

ADOH 利用 AlexNet 网络生成采样数据集的浮点高维特征, 并在之后将全连接层作为哈希层, 将高维浮点特征映射为二值编码, 其形式化定义如式 (1) 所示:

$$B_{q_j}^t = B_{q_j}^t = \text{sign}(f(x_{q_j}^t; \theta)), j = 1, 2, \dots, d \quad (1)$$

式中, θ 为网络的参数; $f(\cdot)$ 为哈希层的输出。

为了保证所生成的二值编码的近邻检索性能较优, ADOH 算法设计了汉明语义一致性模块、单样本语义一致性模块、编码一致模块和编码松弛模块, 下面将分别阐述。

1) 汉明语义一致性模块。为了在汉明空间内得到较优的近邻检索性能, 应在汉明空间内保持样本对之间的原语义相似性关系。如果两个样本在原语义空间中相似, 那么它们在汉明空间内应具有相似的哈希码, 反之亦然。为此, ADOH 算法建立了汉明语义一致性损失函数, 通过最小化哈希码内积与相似度矩阵之间的差异^[11, 27-28], 保持采样数据集与新、旧数据集在语义空间和汉明空间内具有一致的相似性关系, 其形式化定义如式 (2) 所示:

$$\min_{\mathbf{B}'_s, \theta} \left\| \mathbf{B}'_e{}^T \text{sign}(f(X'_q; \theta)) - k \mathbf{S}'_{\text{eq}} \right\|_F^2 + \left\| \mathbf{B}'_s{}^T \text{sign}(f(X'_q; \theta)) - k \mathbf{S}'_{\text{sq}} \right\|_F^2 \quad (2)$$

式中, $\|\cdot\|_F^2$ 为 Frobenius 范数; \mathbf{S}'_{sq} 表示新数据集和采样数据集样本之间的语义相似性矩阵; \mathbf{S}'_{eq} 表示旧数据集和采样数据集样本之间的语义相似性矩阵。语义相似性矩阵可根据样本对的标签生成。若 x'_{s_i} 和 x'_{q_j} 的标签 l'_{s_i} 和 l'_{q_j} 相同, 则 $S'_{s_i q_j} = 1$, 反之, $S'_{s_i q_j} = -1$ 。

2) 单样本语义一致性模块。ADOH 算法为了学习采样数据集单个样本哈希码的语义信息, 在哈希网络层后建立了一个全连接语义网络层, 将 softmax 作为激活函数, 计算样本哈希码的语义标签概率分布, 其定义如式 (3) 所示:

$$g'_{q_j} = \text{softmax}(\mathbf{W}'_s \text{sign}(f(X'_q; \theta)) + \mathbf{V}'_s) \quad (3)$$

式中, $\mathbf{W}'_s \in \mathbb{R}^{c \times k}$ 表示第 t 批次语义层的权重矩阵; $\mathbf{V}'_s \in \mathbb{R}^{c \times 1}$ 表示第 t 批次语义层的偏差向量。

进一步, ADOH 算法定义了分类损失项, 通过最小化交叉熵损失, 使所生成的样本哈希码的语义标签与样本真实标签保持一致, 其形式化定义如式 (4) 所示:

$$\min_{\mathbf{B}'_s, \theta} -\delta \left(\sum_{j \in (\tau \cup \pi)} l'_{q_j} \log g'_{q_j} \right) \quad (4)$$

式中, $\theta = \{\theta, \mathbf{W}'_s, \mathbf{V}'_s\}$; δ 为平衡语义损失的超参数; $\phi = \{1, 2, 3, \dots, m_t\}$ 、 $\varsigma = \{1, 2, 3, \dots, n - m_t\}$ 分别为新、旧数据集的样本索引; $\tau = \{i_1, i_2, \dots, i_{d'}\} \in \phi$, $\pi = \{j_1, j_2, \dots, j_{d^*}\} \in \varsigma$ 分别为采样数据集中新、旧样本的索引, $d' + d^* = d$ 。

3) 编码一致模块。为保证同一个样本在采样数据集和新、旧数据集中的编码相同, ADOH 算法要求最小化采样数据集和新、旧数据集哈希码之间的误差, 如式 (5) 所示:

$$\min_{\mathbf{B}'_s, \theta} \mu \left(\sum_{a \in \tau} (\text{sign}(f(x'_{q_a}; \theta)) - \widetilde{\mathbf{B}}'_{s_a})^2 + \sum_{a \in \pi} (\text{sign}(f(x'_{q_a}; \theta)) - \widetilde{\mathbf{B}}'_{e_a})^2 \right) \quad (5)$$

式中, μ 为平衡参数。若 $a \in \tau$, $\mathbf{B}'_{q_a} = \text{sign}(f(x'_{q_a}; \theta))$ 表示采样数据 x'_{q_a} 通过深度哈希网络生成的哈希码, $\widetilde{\mathbf{B}}'_{s_a}$ 表示新数据 \widetilde{x}'_{s_a} 待学习的哈希码。若 $a \in \pi$ 时, $\mathbf{B}'_{q_a} = \text{sign}(f(x'_{q_a}; \theta))$ 表示采样数据 x'_{q_a} 通过深度哈希网络生成的哈希码, $\widetilde{\mathbf{B}}'_{e_a}$ 表示旧数据 \widetilde{x}'_{e_a} 的哈希码。

4) 编码松弛模块。由于式 (2)、式 (3) 和式 (5) 中包含 $\text{sign}(\cdot)$ 函数, 而 $\text{sign}(\cdot)$ 函数会导致优化过程中存在 NP-hard 问题。为此, ADOH 算法对编码过程进行连续松弛处理, 利用 $\tanh(\cdot)$ 函数代替 $\text{sign}(\cdot)$ 函数。结合式 (2)、式 (4) 和式 (5), 非对称深度在线哈希网络的目标函数定义如式 (6) 所示:

$$L_1 = \min_{\mathbf{B}'_s, \theta} \left\| \mathbf{B}'_e{}^T \tanh(f(X'_q; \theta)) - k \mathbf{S}'_{\text{eq}} \right\|_F^2 + \left\| \mathbf{B}'_s{}^T \tanh(f(X'_q; \theta)) - k \mathbf{S}'_{\text{sq}} \right\|_F^2 + \mu \left(\sum_{a \in \tau} (\tanh(f(x'_{q_a}; \theta)) - \widetilde{\mathbf{B}}'_{s_a})^2 + \sum_{a \in \pi} (\tanh(f(x'_{q_a}; \theta)) - \widetilde{\mathbf{B}}'_{e_a})^2 \right) - \delta \left(\sum_{j \in (\tau \cup \pi)} l'_{q_j} \log g'_{q_j} \right) + \gamma \left\| \tanh(f(X'_q; \theta)) \mathbf{1} \right\|_F^2 \quad (6)$$

式中, $\left\| \tanh(f(X'_q; \theta)) \mathbf{1} \right\|_F^2$ 为平衡项; γ 为平衡项的超参数; $\mathbf{1}$ 为全 1 向量。 $\left\| \tanh(f(X'_q; \theta)) \mathbf{1} \right\|_F^2$ 用来平衡深度哈希网络所生成的哈希码 “+1” 与 “-1” 之间的数量, 使其数量保持相似^[29]。

2.3 标签嵌入

在非对称深度在线哈希网络训练阶段, 单样本语义一致模块只关注保持采样数据集中少量新、旧数据集的语义信息。为了进一步确保所有新、旧数据集的哈希码能够保持样本自身的语义, ADOH 算法建立语义投影损失函数, 将新、旧数据集所有样本标签投影到对应的哈希码, 要求能够最小化样本哈希码与语义标签投影之间的误差, 形式化定义如式 (7) 所示:

$$L_2 = \min_{\mathbf{B}'_s, \mathbf{P}'} \left\| \mathbf{B}'_s - \mathbf{P}' \mathbf{L}'_s \right\|_F^2 + \left\| \mathbf{B}'_e - \mathbf{P}' \mathbf{L}'_e \right\|_F^2 + \eta \left\| \mathbf{P}' \right\|_F^2 \quad (7)$$

式中, $\mathbf{P}^l \in R^{k \times c}$ 为投影矩阵, 将语义标签投影为哈希码; η 为平衡参数。

综上, 结合式 (6) 和式 (7), 非对称深度在线哈希算法的总目标函数如式 (8) 所示:

$$L = \alpha L_1 + L_2 \quad (8)$$

式中, α 为平衡参数。

2.4 交替优化

ADOH 算法采用交替优化策略, 迭代更新参数, 具体步骤如下。

1) 更新参数 θ

更新参数 $\theta = \{\theta, \mathbf{W}^l, \mathbf{V}_s^l\}$ 时, 可去掉目标函数 L 中不含有参数 θ 的项, 相应的优化目标函数如式 (9) 所示:

$$\begin{aligned} & \min_{\mathbf{B}_s, \theta} \sum_{i \in \mathcal{C}} \sum_{j \in (\tau \cup \pi)} (\mathbf{B}_{e_i}^T \mathbf{H}_j - k \mathbf{S}_{e_i q_j}^l)^2 + \\ & \sum_{i \in \phi} \sum_{j \in (\tau \cup \pi)} (\mathbf{B}_{s_i}^l \mathbf{H}_j - k \mathbf{S}_{s_i q_j}^l)^2 + \\ & \mu \left(\sum_{a \in \tau} (\bar{\mathbf{H}} - \bar{\mathbf{B}}_{s_a}^l)^2 + \sum_{a \in \pi} (\bar{\mathbf{H}} - \bar{\mathbf{B}}_{e_a}^l)^2 \right) - \\ & \delta \left(\sum_{j \in (\tau \cup \pi)} l_{q_j}^l \log g_{q_j}^l \right) + \gamma \sum_{j \in (\tau \cup \pi)} (\mathbf{H}_j \mathbf{1})^2 \end{aligned} \quad (9)$$

式中, \mathbf{H}_j 、 $\bar{\mathbf{H}}$ 和 $\bar{\mathbf{H}}$ 的具体定义如式 (10)~式 (12) 所示:

$$\mathbf{H}_j = \tanh(f(x_{q_j}^l; \theta)) \quad (10)$$

$$\bar{\mathbf{H}} = \tanh(f(\bar{x}_{q_a}^l; \theta)) \quad a \in \tau \quad (11)$$

$$\bar{\mathbf{H}} = \tanh(f(\bar{x}_{q_a}^l; \theta)) \quad a \in \pi \quad (12)$$

更新 θ 时, 固定 \mathbf{B}_s^l 和 \mathbf{P}^l , 在反向传播过程中, 利用梯度下降算法优化参数 θ , 其形式化定义如式 (13) 所示:

$$\begin{aligned} & 2 \left(((\mathbf{B}_e^l)^T \mathbf{H}_j - k \mathbf{S}_{e q}^l) \mathbf{B}_e^l + ((\mathbf{B}_s^l)^T \mathbf{H}_j - k \mathbf{S}_{s q}^l) \mathbf{B}_s^l \right) + \\ & \mu \left(\sum_{a \in \tau} (\bar{\mathbf{B}}_{s_a}^l - \bar{\mathbf{H}}_j) + \sum_{a \in \pi} (\bar{\mathbf{B}}_{e_a}^l - \bar{\mathbf{H}}_j) \right) + \gamma \mathbf{H}_j \cdot \\ & (\mathbf{1} - \mathbf{H}_j \cdot \mathbf{H}_j) + \delta (-\mathbf{W}_s^l (l_{q_j}^l - g_{q_j}^l)) \end{aligned} \quad (13)$$

2) 更新 \mathbf{P}^l

更新参数 \mathbf{P}^l 时, 固定参数 θ 和 \mathbf{B}_s^l , 目标函数 L 去掉无关项后并对 \mathbf{P}^l 求导, 如式 (14) 所示:

$$(\mathbf{B}_e^l (\mathbf{L}_e^l)^T + \mathbf{B}_s^l (\mathbf{L}_s^l)^T) \times (\mathbf{L}_s^l (\mathbf{L}_s^l)^T + \mathbf{L}_e^l (\mathbf{L}_e^l)^T + \eta \mathbf{I})^{-1} \quad (14)$$

式中, $\mathbf{I} \in R^{c \times c}$ 为单位矩阵。

3) 更新 \mathbf{B}_s^l

更新参数 \mathbf{B}_s^l 时, 固定参数 \mathbf{P}^l 和 θ , 对目标函数 L 去掉无关项后, 相应目标函数的定义如式 (15) 所示:

$$\begin{aligned} & \min_{\mathbf{B}_s^l} \|\mathbf{B}_s^l \mathbf{B}_q^l - k \mathbf{S}_{s q}^l\|_F^2 + \mu \|\mathbf{V}_q^l - \bar{\mathbf{B}}_s^l\|_F^2 + \|\mathbf{B}_s^l - \mathbf{P}^l \mathbf{L}_s^l\|_F^2 = \\ & \|\mathbf{B}_s^l \mathbf{B}_q^l\|_F^2 - 2k \text{tr}(\mathbf{B}_s^l \mathbf{S}_{s q}^l \mathbf{B}_q^l) - \\ & 2\mu \text{tr}(\bar{\mathbf{B}}_s^l \mathbf{V}_q^l) - 2 \text{tr}(\mathbf{B}_s^l \mathbf{P}^l \mathbf{L}_s^l) \end{aligned} \quad (15)$$

式中, $\mathbf{B}_q^l \in \{+1, -1\}^{k \times d}$ 为采样数据集的哈希码; $\bar{\mathbf{X}}_s^l$ 为新数据集的采样样本。 $\mathbf{V}_q^l = \{v_i | i \in \tau\} \in \{+1, -1\}^{k \times d'}$ 表示 $\bar{\mathbf{X}}_s^l$ 非对称深度网络生成的哈希码, $\bar{\mathbf{B}}_s^l \in \{+1, -1\}^{k \times d'}$ 表示本阶段 $\bar{\mathbf{X}}_s^l$ 待学习的哈希码。 $\phi = \{1, 2, \dots, m_l\}$ 为新数据集中的样本索引。 $\tau = \{i_1, i_2, \dots, i_{d'}\} \in \phi$ 为新数据集采样样本的索引, $\mathbf{V}_q^l \subset \mathbf{B}_q^l$ 。 $\text{tr}(\cdot)$ 表示矩阵迹运算。

在式 (15) 中, \mathbf{V}_q^l 仅为新数据集的采样样本哈希码; \mathbf{B}_s^l 为全部新数据集的哈希码。为了保证 \mathbf{V}_q^l 和 \mathbf{B}_s^l 的维度一致, 重新定义了 v_i , 如式 (16) 所示:

$$\bar{v}_i = \begin{cases} v_i, & i \in \tau \\ 0, & \text{其他} \end{cases} \quad (16)$$

令 $\bar{\mathbf{V}}_q^l = \{\bar{v}_i | i \in \phi\} \in \{+1, -1\}^{k \times m_l}$, 则式 (15) 可重新定义为式 (17):

$$\begin{aligned} & \min_{\mathbf{B}_s^l} \|\mathbf{B}_s^l \mathbf{B}_q^l\|_F^2 - 2 \text{tr}(\mathbf{B}_s^l (k \mathbf{V}_q^l \mathbf{S}_{s q}^l + \mu \bar{\mathbf{V}}_q^l + \mathbf{P}^l \mathbf{L}_s^l)) = \\ & \min_{\mathbf{B}_s^l} \|\mathbf{B}_s^l \mathbf{B}_q^l\|_F^2 + \text{tr}(\mathbf{B}_s^l \mathbf{C}) \end{aligned} \quad (17)$$

式中, $\mathbf{C} = -2(k \mathbf{B}_q^l \mathbf{S}_{s q}^l + \mu \bar{\mathbf{V}}_q^l + \mathbf{P}^l \mathbf{L}_s^l)$ 。

根据式 (17) 更新 \mathbf{B}_s^l 时, 采用离散循环坐标法^[3] 对 \mathbf{B}_s^l 逐行求解, 则其第 l 行的优化目标函数如式 (18) 所示:

$$\min_{\mathbf{B}_{s_l}^l} \text{tr}(\mathbf{B}_{s_l}^l (2 \bar{\mathbf{B}}_{s_l}^l \bar{\mathbf{B}}_{q_l}^l \mathbf{B}_{q_l}^l + \mathbf{C}_l)) \quad (18)$$

式中, $\mathbf{B}_{s_l}^l$ 、 $\mathbf{B}_{q_l}^l$ 、 \mathbf{P}_l 分别表示 \mathbf{B}_s^l 、 \mathbf{B}_q^l 、 \mathbf{C} 的第 l 行; $\bar{\mathbf{B}}_{s_l}^l$ 、 $\bar{\mathbf{B}}_{q_l}^l$ 、 $\bar{\mathbf{C}}_l$ 分别表示 \mathbf{B}_s^l 、 \mathbf{B}_q^l 、 \mathbf{C} 除了第 l 行之外的其他行。

根据式 (18), $\mathbf{B}_{s_l}^l$ 的计算方式如式 (19) 所示:

$$\mathbf{B}_{s_l}^l = -\text{sign}(2 \bar{\mathbf{B}}_{s_l}^l \bar{\mathbf{B}}_{q_l}^l \mathbf{B}_{q_l}^l + \mathbf{C}_l) \quad (19)$$

综上, 非对称深度在线哈希算法的训练过程如下述算法所示。

输入: 新、旧数据集图像标签: $\mathbf{L} = \{\mathbf{L}_s^l, \mathbf{L}_e^l\}$

新、旧数据集图像数据: $\mathbf{X} = \{\mathbf{X}_s^l, \mathbf{X}_e^l\}^n$

旧数据图像数据集生成的哈希码: \mathbf{B}_e^l

新数据集和采样数据集标签相似性矩阵: $\mathbf{S}_{s q}^l$

旧数据集和采样数据集标签相似性矩阵: $\mathbf{S}_{e q}^l$

设置新数据最大流入批次 T ;

设置网络最大迭代次数 Z ;

设置哈希码长度 k ;
 初始化新数据哈希码和网络参数;
 运行 $t=1 \rightarrow T$
 更新 \mathbf{B}'_e 和 \mathbf{X}'_e
 运行 $i=1 \rightarrow Z$
 根据式 (13) 计算梯度;
 利用反向传播算法更新网络参数 $\theta = \{\theta, \mathbf{W}'_s, \mathbf{V}'_s\}$;
 满足迭代条件后结束;
 根据式 (14) 更新 \mathbf{P}' ;
 根据式 (19) 更新 \mathbf{B}'_s ;
 满足迭代条件后结束;
 输出: 新数据集哈希码 \mathbf{B}'_s

2.5 时间复杂度分析

ADOH 算法的训练过程主要包括迭代更新参数 θ 、 \mathbf{P}' 和 \mathbf{B}'_s 。在本文中, n 表示新、旧数据集中样本数量, n_t 表示新数据集样本数量, d 表示采样数据集样本数量, c 表示数据集类别, k 表示哈希码长度, 算法迭代次数为 T , 则更新 θ 的时间复杂度为 $O(ndk+cd)$, 更新 \mathbf{P}' 的时间复杂度为 $O(kcn+c^2n+c^3)$, 更新 \mathbf{B}'_s 的时间复杂度为 $O(ndk^2)$, 算法的总时间复杂度为 $O((dk+kc+c^2)n+dk^2n_t)$ 。由于 c 、 k 和 d 的值远远小于 n 和 n_t , 所以 ADOH 算法时间复杂度最终表示为 $O(T(n+n_t))$ 。ADOH 算法的时间复杂度与样本数量呈线性关系, 与基于可扩展监督在线哈希^[1]和基于平衡相似性的标签投影在线哈希^[2]的时间复杂度相同, 略高于类智能快速在线哈希算法^[3]时间复杂度 ($O(Tq)$, q 表示数据的维度), 但本文所提出的 ADOH 算法的近邻检索性能最优。

3 实验

3.1 数据集

在本文中, 选取 3 个广泛使用的数据集 cifar-10 数据集^[30]、mnist 数据集^[31] 和 Places205 数据集^[32]。

cifar-10 数据集包含 60 000 张图像, 共分为 10 个类别, 每个类别含有 6 000 张图像, 图像大小为 32×32 。根据 BSODH^[11]、FCOH^[13] 等在线哈希算法, 随机选取 59 000 张图像为检索集, 1 000 张图像为测试集。同时, 将检索集被分为新、旧数据集, 新数据集每批次含有 2 000 张图像, 共 10 批次。

mnist 数据集含有 70 000 张手写数字图像, 共分为 10 个类, 图像大小为 28×28 。类似地, 每个类别随机选取 100 张图像为测试集, 其余为检索集。同时, 将检索集划为新、旧数据集, 新数据集每批次包含 2 000 张图像, 共 10 批次。

Places205 数据集为大型场景数据集, 包含 205 个场景类别, 共 250 万张图像, 每张图像的大小为 256×256 。本次实验从 Places205 数据集中随机抽取 35 个场景类别, 每个类别随机选取 500~2 000 张图像, 共 45 059 张图像。从每个类别随机抽取 20 张图像作为测试集, 其余图像作为检索集。检索集被分为旧数据集和新数据集, 其中新数据分为 10 个批次, 每个批次包含 2 000 张图像。

3.2 基准算法和评价指标

基准对比算法包括 OKH^[22]、AdaptHash^[23]、OSH^[7]、MIHash^[24]、BSODH^[11]、HCOH^[8]、HMOH^[9] 和 FCOH^[13]。为了有效评估近邻检索性能, 选取了 3 个指标: mAP、Precision@H2 和 Precision@K。

mAP 表示平均精确度均值, 给定一个查询样本 q , 其平均精确度均值的定义如式 (20) 所示:

$$\text{mAP} = \frac{1}{Q} \frac{1}{L_q} \sum_{r=1}^Q \sum_{r=1}^n P_q(r) \delta_q(r) \quad (20)$$

式中, Q 为检索样本的数量; n 为返回样本的个数; L_q 为检索正确样本的个数; $P_q(r)$ 为检索到第 r 个实例的精度, 如果第 r 个样本是真实近邻点, 则 $\delta_q(r)=1$, 否则, $\delta_q(r)=0$ 。

Precision 为准确率, 也可称为查准率, 表示所有检索返回样本中正确样本所占的百分比。Precision@H2 表示以查询点为中心、半径为 2 的汉明球内的准确率; Precision@K 表示检测到前 K 个近邻点的准确率, $K=[1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100]$ 。如图 2 所示, 测试集样本 A 的哈希码为 100011, 数据库样本 B 的哈希码为 100110, 数据库样本 C 的哈希码为 000110。A 与 B 哈希码之间的汉明距离为 2, A 与 B 哈希码之间的汉明距离为 3。则测试集样本 A 的 Precision@H2 结果样本集合包含数据库样本 B, 不包含数据库样本 C。但在计算 Precision@K($K=2$) 时, 样本 B、C 将均作为检索结果用于计算查找准确率。

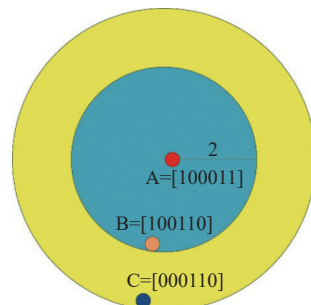


图 2 样本之间汉明距离示意图

3.3 在线近邻检索性能

在线近邻检索实验中, 哈希码长度为 16、32、48、64 和 128 bits。实验结果如表 2~表 4 和图 3~图 5 所示。

1) mAP 和 Precision@H2

在 cifar-10 数据集上, 本算法与基准算法在不同哈希码长度下的 mAP 和 Precision@H2 结果如表 2 所示。由 mAP 评估指标结果可知, 当哈希码长度为 16 和 32 bits 时, ADOH 算法比次优算法

HCOH 分别提高了 15.0% 和 16.2%; 当哈希码长度为 48 bits 时, ADOH 算法比次优算法 HMOH 提高了 17.2%, 当哈希码长度为 64 和 128 bits 时, ADOH 算法比次优算法 FCOH 分别提高了 15.3% 和 14.5%。由 Precision@H2 评估指标结果可知, 当哈希码长度为 16、32 和 48 bits 时, ADOH 算法相较于次优算法 FCOH 分别提高了 9.2%, 5.8% 和 7.8%; 当哈希码长度为 64 和 128 bits 时, ADOH 算法相较于次优算法 HMOH 分别提高了 5.7% 和 2.5%。

表 2 cifar-10 数据集上不同长度哈希码的 mAP 和 Precision@H2 结果对比

| 算法 | mAP | | | | | Precision@H2 | | | | |
|-------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | 16 bits | 32 bits | 48 bits | 64 bits | 128 bits | 16 bits | 32 bits | 48 bits | 64 bits | 128 bits |
| OKH | 0.063 | 0.166 | 0.298 | 0.330 | 0.356 | 0.100 | 0.184 | 0.433 | 0.334 | 0.101 |
| AdaptHash | 0.255 | 0.216 | 0.196 | 0.198 | 0.219 | 0.270 | 0.341 | 0.388 | 0.395 | 0.322 |
| OSH | 0.124 | 0.116 | 0.122 | 0.131 | 0.135 | 0.581 | 0.184 | 0.100 | 0.082 | 0.069 |
| MIHash | 0.638 | 0.630 | 0.614 | 0.601 | 0.563 | 0.605 | 0.573 | 0.649 | 0.600 | 0.496 |
| BSODH | 0.640 | 0.690 | 0.675 | 0.693 | 0.699 | 0.642 | 0.690 | 0.687 | 0.672 | 0.542 |
| HMOH | 0.710 | 0.718 | 0.721 | 0.732 | 0.735 | 0.736 | 0.741 | 0.704 | 0.715 | 0.722 |
| HCOH | 0.736 | 0.732 | 0.717 | 0.726 | 0.734 | 0.728 | 0.734 | 0.695 | 0.607 | 0.497 |
| FCOH | 0.674 | 0.702 | 0.711 | 0.739 | 0.742 | 0.738 | 0.743 | 0.711 | 0.648 | 0.618 |
| ADOH | 0.886 | 0.894 | 0.893 | 0.892 | 0.887 | 0.830 | 0.801 | 0.789 | 0.772 | 0.747 |

表 3 mnist 数据集上不同长度哈希码的 mAP 和 Precision@H2 结果对比

| 算法 | mAP | | | | | Precision@H2 | | | | |
|-------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | 16 bits | 32 bits | 48 bits | 64 bits | 128 bits | 16 bits | 32 bits | 48 bits | 64 bits | 128 bits |
| OKH | 0.100 | 0.180 | 0.302 | 0.333 | 0.398 | 0.100 | 0.394 | 0.660 | 0.698 | 0.139 |
| AdaptHash | 0.194 | 0.224 | 0.205 | 0.259 | 0.223 | 0.215 | 0.289 | 0.303 | 0.259 | 0.375 |
| OSH | 0.143 | 0.133 | 0.129 | 0.163 | 0.167 | 0.134 | 0.131 | 0.113 | 0.119 | 0.042 |
| MIHash | 0.630 | 0.710 | 0.688 | 0.733 | 0.724 | 0.698 | 0.794 | 0.754 | 0.712 | 0.421 |
| BSODH | 0.659 | 0.710 | 0.723 | 0.739 | 0.739 | 0.694 | 0.795 | 0.785 | 0.723 | 0.532 |
| HMOH | 0.702 | 0.707 | 0.730 | 0.713 | 0.702 | 0.769 | 0.807 | 0.803 | 0.818 | 0.792 |
| HCOH | 0.699 | 0.752 | 0.749 | 0.771 | 0.786 | 0.785 | 0.836 | 0.790 | 0.630 | 0.571 |
| FCOH | 0.725 | 0.786 | 0.789 | 0.784 | 0.801 | 0.817 | 0.849 | 0.814 | 0.817 | 0.620 |
| ADOH | 0.984 | 0.985 | 0.986 | 0.987 | 0.984 | 0.983 | 0.976 | 0.966 | 0.955 | 0.954 |

在 mnist 数据集上, 本算法与基准算法在不同哈希码长度下的 mAP 和 Precision@H2 结果如表 3 所示。由 mAP 评估指标结果可知, 当哈希码长度为 16、32、48、64 和 128 bits 时, ADOH 算法比次优算法 FCOH 分别提高了 25.9%、19.9%、19.7%、20.3% 和 18.3%。由 Precision@H2 评估指标结果可知, 当哈希码长度为 16、32 和 48 bits 时, ADOH 算法比次优算法 FCOH 分别提高了 16.6%、12.7% 和 15.2%; 当哈希码长度为 64 和 128 bits 时, ADOH 算法比次优算法 HMOH 分别提高了 13.7% 和 16.2%。

在 Places205 数据集上, 本算法与基准算法在不同哈希码长度下的 mAP 和 Precision@H2 结果如表 4 所示。由 mAP 评估指标结果可知, ADOH 算法比次优算法 FCOH 分别提高了 4.5%、5.6%、5.7%、4.6% 和 3.1%。在 Precision@H2 评估指标对比实验的结果中, 当哈希码长度为 16、32、48 和 64 bits 时, ADOH 算法比次优算法 FCOH 分别提高了 6.8%、4.2%、3.6% 和 5.6%; 当哈希码长度为 128 bits 时, ADOH 算法比次优算法 HMOH 提高了 6%。

表 4 Places205 数据集上不同长度哈希码的 mAP 和 Precision@H2 结果对比

| 算法 | mAP | | | | | Precision@H2 | | | | |
|-------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | 16 bits | 32 bits | 48 bits | 64 bits | 128 bits | 16 bits | 32 bits | 48 bits | 64 bits | 128 bits |
| OKH | 0.061 | 0.198 | 0.265 | 0.287 | 0.298 | 0.086 | 0.210 | 0.278 | 0.156 | 0.075 |
| AdaptHash | 0.238 | 0.308 | 0.318 | 0.323 | 0.327 | 0.247 | 0.313 | 0.347 | 0.295 | 0.221 |
| OSH | 0.041 | 0.117 | 0.215 | 0.248 | 0.256 | 0.127 | 0.152 | 0.205 | 0.172 | 0.159 |
| MIHash | 0.325 | 0.376 | 0.433 | 0.485 | 0.498 | 0.351 | 0.371 | 0.416 | 0.308 | 0.293 |
| BSODH | 0.431 | 0.465 | 0.480 | 0.531 | 0.575 | 0.447 | 0.478 | 0.492 | 0.455 | 0.384 |
| HMOH | 0.429 | 0.488 | 0.492 | 0.546 | 0.591 | 0.405 | 0.477 | 0.481 | 0.502 | 0.491 |
| HCOH | 0.446 | 0.495 | 0.525 | 0.554 | 0.602 | 0.421 | 0.514 | 0.512 | 0.464 | 0.438 |
| FCOH | 0.467 | 0.507 | 0.574 | 0.587 | 0.605 | 0.451 | 0.526 | 0.588 | 0.557 | 0.479 |
| ADOH | 0.512 | 0.563 | 0.631 | 0.633 | 0.636 | 0.519 | 0.568 | 0.624 | 0.613 | 0.551 |

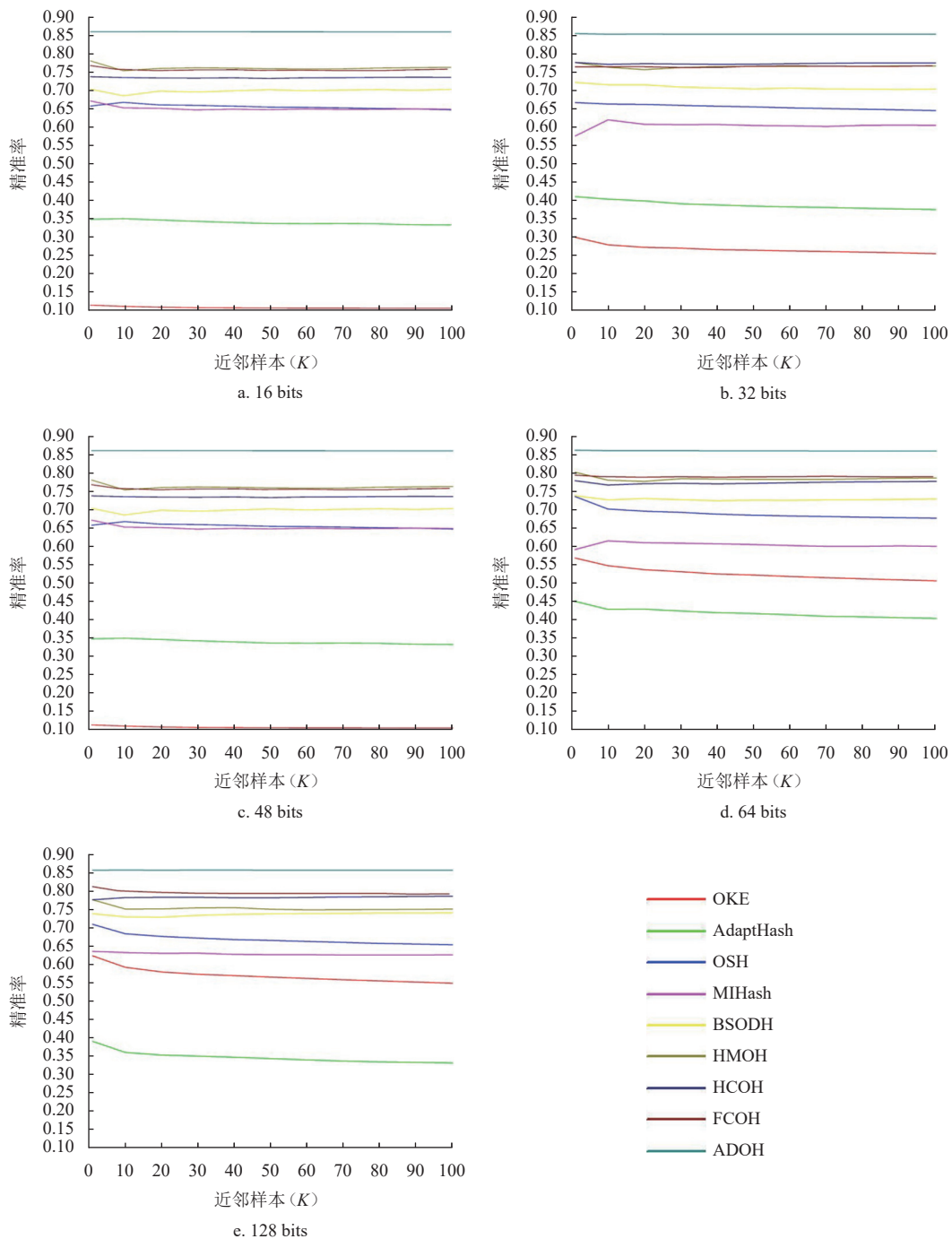


图 3 cifar-10 数据集上不同长度哈希码的 Precision@K 结果对比

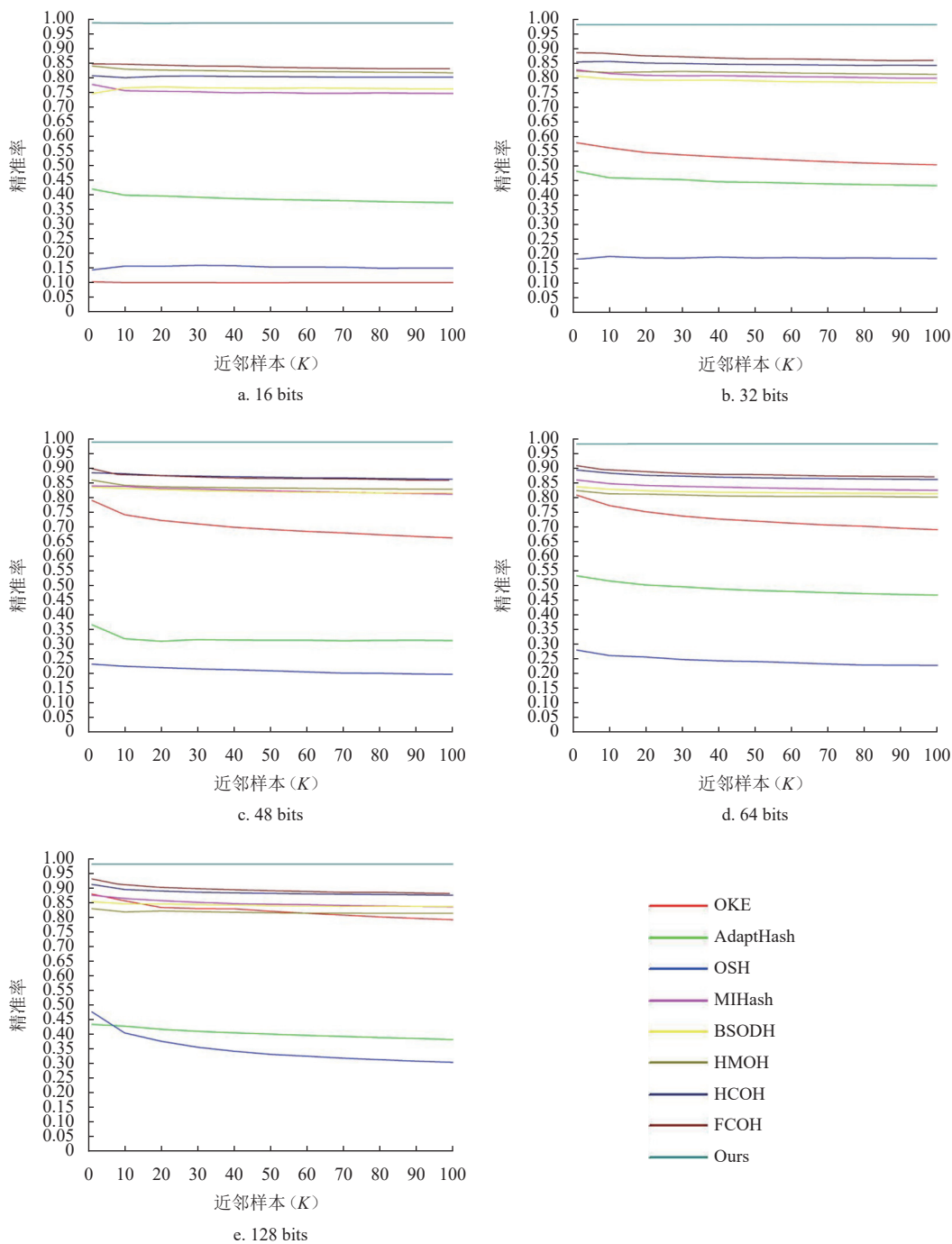


图 4 mnist 数据集上不同长度哈希码的 Precision@K 结果对比

在表 2~表 4 中, 随着哈希码长度的增加, 各个哈希算法的在线近邻检索性能指标 mAP 结果均出现不同程度的波动。但相较而言, ADOH 算法的 mAP 值的波动最小, 近邻检索性能更稳定, 如在 mnist 数据集上 mAP 值的波动仅为 0.3%。

综上, 在数据集 cifar-10、mnist 和 Places205 上, ADOH 算法的 Precision@H2 和 mAP 的结果均

最优。在构建在线哈希模型时, FCOH、BSODH 等算法仅考虑样本之间的相似性关系, HCOH、HMOH 等算法仅考虑单个样本的语义信息, 而 ADOH 算法兼顾样本之间的语义相似性关系和单个样本的语义信息。再者, 与 FCOH、HMOH 等浅模型基准算法不同, ADOH 算法构建了深度语义在线哈希模型。因此, ADOH 算法的近邻检索性能优于其他算法。

2) Precision@K

图 3~图 5 分别展示了本算法与其他基准算法在数据集 cifar-10、mnist 和 Places205 上的 Precision@K 实验结果。Precision@K 评估指标的邻居搜索范围为 1~100，每跨越 10 个邻居验证一次实验结果，每个在线哈希算法分别在数据集 cifar-10、mnist 和 Places205 上进行 11 次实验。从图 3~图 5 可以看出，所提算法的 Precision@K 值高于其

他基准算法。当取不同范围的邻居时，基准算法的实验结果均存在不同程度的波动，而 ADOH 算法在不同邻居取值下 Precision@K 结果几乎未出现波动。相较于其他基准算法，ADOH 算法的检索性能更优、更稳定。再者，哈希码长度较短时，本文所提 ADOH 算法具备优异的近邻检索性能，更容易扩展到大规模图像检索任务中。

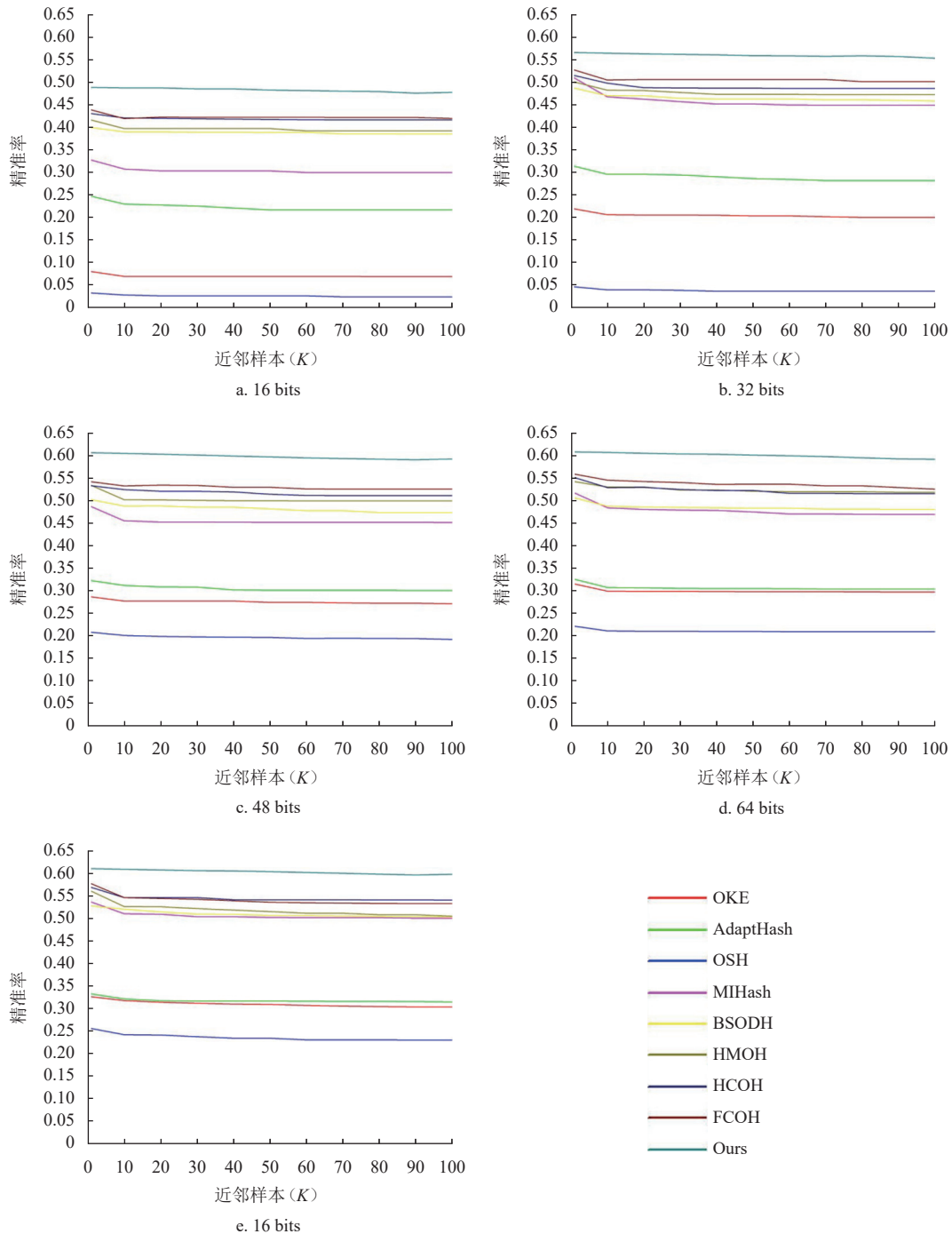


图 5 Places205 数据集上不同长度哈希码的 Precision@K 结果对比

3.4 参数敏感度分析

ADOH 算法的目标函数共包含 5 个调节参数 μ 、 γ 、 δ 、 η 和 α 。为了确定它们的合理取值, 本文选取 Precision@H2 和 mAP 为评估指标, 在 cifar-10 数据集上对比了不同参数值的 ADOH 算法近邻检索性能, 其哈希码长度设置为 16 bits。

1) 参数 α 的影响

α 为平衡目标函数 L 的参数, 平衡非对称深度网络模块和标签嵌入模块的权重。本次实验设置根据深度增量哈希网络^[29] 设置参数 $\mu=200$, $\gamma=50$, $\alpha=[0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8]$, 参数 δ 和 η 的值为 0。在 mAP 和 Precision@H2 评估指标下, 不同参数 α 值的近邻检索性能如图 6 所示。实验结果表明, $\alpha=1$ 时, mAP 和 Precision@H2 的值最大。

从图 6 中可以看出, 当 $\alpha=0.2$, 非对称深度网络模块的权重最低时, ADOH 算法的检索性能最差。非对称深度网络模块利用深度网络提取图像特征, 并包含汉明语义一致性模块和单样本语义一致性模块, 标签嵌入模块仅通过标签投影学习单个样本的语义信息。相比于标签嵌入模块, 非对称深度网络模块更影响算法的性能。

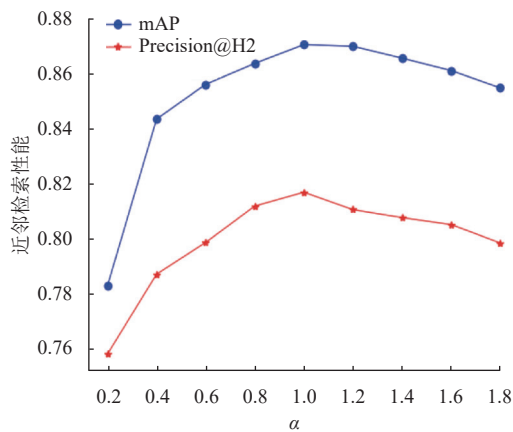


图 6 α 值的实验结果

2) 参数 μ 、 γ 和 δ 的影响

μ 量化哈希码误差的参数, γ 为平衡项的参数, δ 为平衡交叉熵标签语义损失的参数。本次实验设置 $\mu=[1, 10, 100, 200, 500]$, $\gamma=[0.1, 1, 10, 50, 100]$, $\delta=[0.2, 0.4, 0.6, 0.8, 1.0]$ 。

选取参数 μ 时, 设置参数 $\gamma=50$, 参数 δ 和 η 的值为 0, $\alpha=1$ 。由图 7 可知, $\mu=200$ 时, ADOH 算法的近邻检索性能指标 mAP 和 Precision@H2 的值最优。

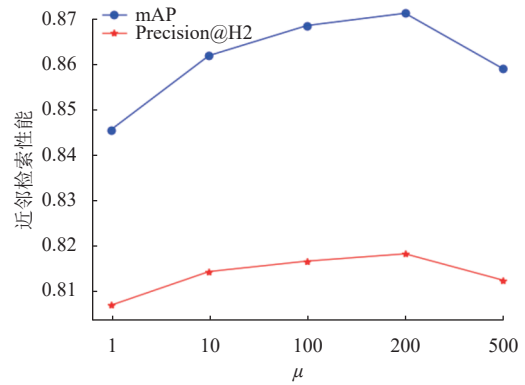


图 7 μ 的实验结果

选取参数 γ 时, 设置参数 $\mu=200$, 参数 δ 和 η 的值为 0, $\alpha=1$ 。由图 8 可知, $\gamma=50$ 时, ADOH 算法的近邻检索性能最优。

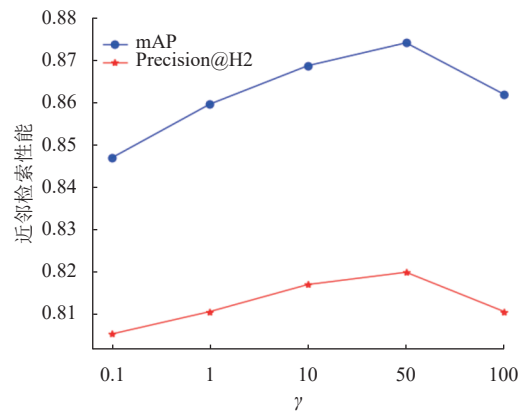


图 8 γ 的实验结果

选取参数 δ 时, 设置参数 $(\mu, \gamma, \eta, \alpha) = (200, 50, 0, 1)$ 。由图 9 可知, $\delta=0.6$ 时, ADOH 算法的近邻检索性能指标 mAP 和 Precision@H2 值最大。

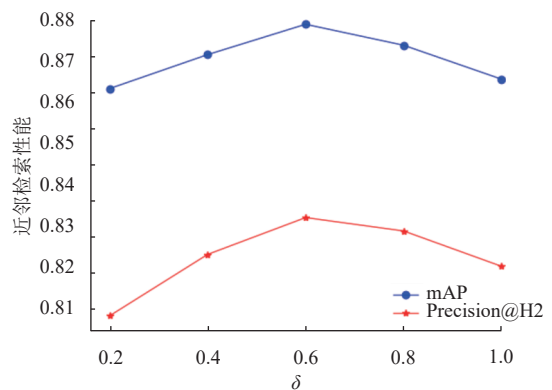


图 9 δ 的实验结果

综上所述, $(200, 50, 0.6)$ 为 μ 、 γ 和 δ 的最佳参数组合。

3) 参数 η 的影响

在标签嵌入模块, 设置了平衡参数 η , 可有效

避免过拟合现象。本次实验设置 η 的取值范围为[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]，并设置参数 $(\mu, \gamma, \delta, \alpha) = (200, 50, 0.6, 1)$ 。图 10 为不同 η 值的近邻检索性能 mAP 和 Precision@H2 指标值，并且 $\eta=0.6$ 时，ADOH 算法的近邻检索性能最好。

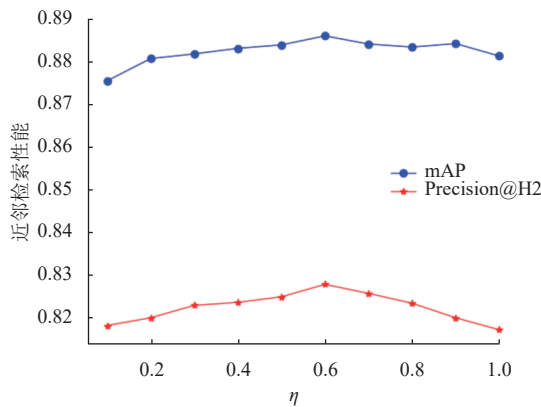


图 10 η 的实验结果

4 结束语

本文提出一种新的深度监督在线哈希算法，称为非对称深度在线哈希算法（ADOH）。ADOH 提出利用深度神经网络学习在线哈希模型，保持新、旧数据集在原始特征空间和语义空间的相似性关系。ADOH 通过非对称学习的方式，只学习新数据集的哈希码，保持旧数据集的哈希码不变，提高模型的学习效率。ADOH 在非对称深度在线哈希网络学习中建立分类损失项，利用交叉熵预测标签，充分挖掘数据点本身的深度语义信息。另外，ADOH 通过标签嵌入模块，分别将新、旧数据集标签投影到对应的哈希码中，使类别相同的数据生成的哈希码趋于一致，使待学习的哈希码更易于分类。在 3 个广泛使用的数据集 cifar-10、mnist 和 Places205 上设置了近邻检索性能对比实验，实验结果表明 ADOH 算法的性能优于目前 8 种较先进的在线哈希算法。

参考文献

- [1] FANG Y Z, ZHANG H X, LIU L. Label projection online Hashing for balanced similarity[J]. *Journal of Visual Communication and Image Representation*, 2021, 80: 103314.
- [2] WANG J, KUMAR S, CHANG S F. Semi-supervised Hashing for scalable image retrieval[C]//Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2010: 3424-3431.
- [3] SHEN F M, SHEN C H, LIU W, et al. Supervised discrete Hashing[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2015: 37-45.
- [4] GUI J, LIU T L, SUN Z N, et al. Fast supervised discrete Hashing[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018, 40(2): 490-496.
- [5] GUI J, LI P. R~2SDH: Robust rotated supervised discrete Hashing[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York: ACM, 2018: 1485-1493.
- [6] ZHU X F, HUANG Z, SHEN H T, et al. Linear cross-modal Hashing for efficient multimedia search[C]//Proceedings of the 21st ACM International Conference on Multimedia. New York: ACM, 2013: 143-152.
- [7] CAKIR F, BARGAL S A, SCLAROFF S. Online supervised Hashing[J]. *Computer Vision and Image Understanding*, 2017, 156: 162-173.
- [8] LIN M B, JI R R, LIU H, et al. Supervised online Hashing via hadamard codebook learning[C]//Proceedings of the 26th ACM International Conference on Multimedia. New York: ACM, 2018: 1635-1643.
- [9] LIN M B, JI R R, LIU H, et al. Hadamard matrix guided online Hashing[J]. *International Journal of Computer Vision*, 2020, 128(8): 2279-2306.
- [10] KITTLER J, GHADERI R, WINDEATT T, et al. Face verification using error correcting output codes[C]//Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2001: 755-760.
- [11] LIN M B, JI R R, LIU H, et al. Towards optimal discrete online Hashing with balanced similarity[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Honolulu: AAAI, 2019: 8722-8729.
- [12] FANG Y Z, LIU L. Scalable supervised online Hashing for image retrieval[J]. *Journal of Computational Design and Engineering*, 2021, 8(5): 1391-1406.
- [13] LIN M B, JI R R, SUN X S, et al. Fast class-wise updating for online Hashing[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022, 44(5): 2453-2467.
- [14] LI W J, WANG S, KANG W C. Feature learning based deep supervised Hashing with pairwise labels [C]. *International Joint Conference on Artificial Intelligence*. Buenos Aires: AAAI Press, 2015: 1711-1717.
- [15] LI Q, SUN Z N, HE R, et al. Deep supervised discrete Hashing[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. New York: ACM, 2017: 2479-2488.
- [16] NGUYEN V A, DO M N. Deep learning based supervised Hashing for efficient image retrieval[C]//Proceedings of the IEEE International Conference on Multimedia and Expo. New York: IEEE, 2016: 1-6.
- [17] JIANG Q Y, LI W J. Asymmetric deep supervised Hashing[C]//Proceedings of the AAAI Conference on Artificial Intelligence. New Orleans: AAAI, 2018: 3342-3349.

- [18] ZHANG M, CHENG C, LONG X Z. Deep semantic asymmetric Hashing[C]//Proceedings of the International Conference on Artificial Neural Networks. Cham: Springer, 2019: 363-374.
- [19] SONG W W, GAO Z, DIAN R W, et al. Asymmetric Hash code learning for remote sensing image retrieval[J]. IEEE Transactions on Geoscience and Remote Sensing, 2022, 60: 1-14.
- [20] LENG C, WU J X, CHENG J, et al. Online sketching Hashing[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2015: 2503-2511.
- [21] CHEN X X, KING I, LYU M R. FROSH: FasteR online sketching Hashing[C]//Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence. Sdney: AVAI, 2017: 11093721.
- [22] HUANG L K, YANG Q, ZHENG W S. Online Hashing[C]//Proceedings of the 23rd International Joint Conference on Artificial Intelligence. Beijing: IJCAI, 2013: 1422-1428.
- [23] CAKIR F, SCLAROFF S. Adaptive Hashing for fast similarity search[C]//Proceedings of the IEEE International Conference on Computer Vision. New York: IEEE, 2015: 1044-1052.
- [24] CAKIR F, HE K, BARGAL S A, et al. MIHash: Online Hashing with mutual information[C]//Proceedings of the IEEE International Conference on Computer Vision. New York: IEEE, 2017: 437-445.
- [25] WENG Z Y, ZHU Y S. Online Hashing with efficient updating of binary codes[C]//Proceedings of the 34th Conference on Artificial Intelligence. New York: AAAI, 2020: 12354-12361.
- [26] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks[J]. *Communications of the ACM*, 2017, 60(6): 84-90.
- [27] WU X M, LUO X, ZHAN Y W, et al. Online enhanced semantic Hashing: towards effective and efficient retrieval for streaming multi-modal data[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto: AAAI, 2022: 4263-4271.
- [28] WANG Y X, LUO X, XU X S. Label embedding online Hashing for cross-modal retrieval[C]//Proceedings of the 28th ACM International Conference on Multimedia. New York: ACM, 2020: 871-879.
- [29] WU D Y, DAI Q, LIU J, et al. Deep incremental Hashing network for efficient image retrieval[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2019: 9069-9077.
- [30] KRIZHEVSKY A, HINTON G. Learning multiple layers of features from tiny images[EB/OL]. [2023-06-01]. <https://doi.org/10.1.1.222.9220>.
- [31] LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition[J]. *Proceedings of the IEEE*, 1998, 86(11): 2278-2324.
- [32] ZHOU B L, LAPEDRIZA A, XIAO J X, et al. Learning deep features for scene recognition using places database[J]. *Advances in Neural Information Processing Systems*, 2014, 1: 487-495.

编辑 张莉