Deep Hashing with Minimal-Distance-Separated Hash Centers

Liangdao Wang, Yan Pan,Cong Liu, Hanjiang Lai, Jian Yin, Ye Liu1

摘要

深度哈希是一种具有很大优势的大规模图像检索方法。然而，目前大多数监督深度哈希方法会在随机抽样的小批量中使用成对或三元组图像的相似性来学习哈希函数。这样的方法存在训练效率低、数据分布覆盖不足、配对不平衡等问题。最近，中心相似度量化（CSQ）类的方法利用“哈希中心”作为全局相似度度量来解决上述问题，这鼓励了相似图像的哈希码接近属于他们同一类别的哈希中心，并与其他哈希中心保持距离。这样的方法虽然得到了很好的检索性能，但由于该方法缺少对最小距离的约束，即哈希中心之间的距离可以任意接近。为了解决上述问题，本篇论文提出了（1）一种约束哈希中心最小距离的方法，并针对这个非凸的非平凡优化问题给出了求解方法。(2) 采用了编码理论里的 Gilbert-Varshamov 界限，在获得一个较大的最小距离的同时确保优化方案的可行性。(3) 在获得清晰分离的高质量哈希中心之后，每个中心都被分配给一个图像的类别，设计了一个深度哈希网络并通过几个有效的损失函数进行训练。在三个图像检索数据集上进行的大量实验表明，该方法比现有的深度哈希方法具有更好的检索性能。

关键词： 哈希；深度学习；

1 引言

近年来，由于数据的快速爆炸，检索任务将耗费大量的时间和内存。哈希方法作为一种大规模搜索的强大技术，通过哈希函数将高维数据转换为简单的二进制代码，这样简单而短小的二进制代码可以以低存储成本实现高效的检索速度。因此哈希学习已被广泛应用于机器学习、计算机视觉和图像检索领域。

哈希方法一般分为有监督哈希方法和无监督哈希方法。在早期，通过随机投影生成哈希码的数据独立方法比较流行，具有代表性的有局部敏感哈希（LSH）[1] [2]，该方法也有许多变体，如减少哈希码位数的 LSH 方法 [3]、核局部敏感哈希（KLSH）[4] 等。然而，这些方法只有在学习较长的编码时才能获得更好的检索性能，这导致了巨大的存储和计算成本。为了提高检索准确性，现在更多是采用数据依赖的哈希方法，可以分为无监督哈希方法和有监督哈希方法。无监督哈希方法仅通过数据的特征信息来学习哈希函数，著名的方法有谱哈希（SH）[5]、迭代量化哈希（ITQ）[6] 和锚点图哈希 [7] 等。有监督哈希方法则是利用标签信息构建哈希函数，如最小损失哈希（MLH）[8]、基于核函数的监督哈希（KSH）[9]、监督离散哈希（SDH）[10] 等，提升了哈希方法的检索性能。

而随着深度学习技术的发展，近年来，基于深度学习的哈希（深度哈希）图像检索方法得到了大量的研究。基于卷积神经网络 (CNN) 和深度残差神经网络 (ResNets) 等相关方法的研究表明，深度学习可以有效地刻画数据的非线性结构，大幅提升图像识别率。为了充分利用神经网络的特征学习能力，人们对深度哈希方法进行了大量研究，使用深度神经网络学习哈希函数，将相似/不相似的图像分别编码为相邻/远离的二进制码。大多数现有的深度哈希方法在随机采样的小批量训练样本中训练模型，基于样本之间的成对或三元组相似性（例如 [11] [12] [13] [14]。然而，袁等人 [15] 指出，这些方法存在三个问题，从而导致性能受限：(1) 获取数据集的全局相似性的效率较低：对大规模数据集来说，在 n 个数据点上计算成对的相似性会有 $O(n!)$ 的时间复杂度。因此，对于大规模图像或视频数据来说，对所有可能的数据对/三元组中进行穷尽学习是不切实际的。(2) 数据分布覆盖不足：基于成对/三元组相似性的方法仅利用数据对之间的部分关系，无法对全局的信息进行学习，这可能会损害生成的哈希码的可区分性。(3) 存在不平衡数据的问题：在实际场景中，不相似的数据对的数量要远远大于相似的数据对。因此，基于成对/三元组相似性的哈希方法无法充分学习相似关系从而生成足够好的哈希码，导致性能受限。

为了解决上述问题，袁等人 [15] 提出了中心相似度度量（CSQ）的方法，重新思考了如何设计全局数据相似性建模，提出了一种更有效捕捉数据关系的哈希中心的新概念，为每个类别的相似图像找到相互分离的哈希中心，并使用这些中心来确保相似图像的哈希码之间的距离较小，而不相似图像的哈希码之间的距离较大，其优势在于：（1）较高训练效率。(2) 能够学习数据的全局相似性分布。(3) 没有“成对不平衡”问题。对于使用哈希中心的深度哈希方法来说，构建清晰可分的哈希中心是至关重要的，举例而言，两个哈希中心之间的汉明距离应该显著大于两个相似图像的哈希码之间的汉明距离，但这样的约束使得该算法在泛化到不同长度的哈希代码和不同数量的图像类别任务时具有较大的挑战性。例如，CSQ 采用 Hadamard 矩阵和伯努利采样这两种方法来产生具有良好性质的哈希中心，生成哈希中心的期望是任意两个哈希中心的汉明距离平均是哈希码长度的一半。然而，以这种方式构造的哈希中心在最坏的情况下可以任意小，即汉明距离可能为零。这些退化的哈希中心会损害生成哈希码的性能。

为了解决这个问题，本文提出了一种新颖的深度哈希方法，使用优化过程生成哈希中心，并对给定的任意两个哈希中心之间的最小距离 d 添加了额外约束。作者使用从编码理论中采用的 Gilbert-Varshamov 界限来导出 d 的值，这有助于找到一个较大的 d，同时确保优化过程的可行性。在第一阶段，作者解决上述优化问题，产生明显分离的哈希中心。为了解决这个优化问题，提出了一种交替优化过程，依赖于 $\ell p-box$ 的二进制优化技术 [16]。在第二阶段，使用构建的哈希中心训练深度哈希网络，作为全局相似度度量。具体而言，损失函数被定义为：（1）使得输入图像的哈希码接近其类别的哈希中心，但与其他中心距离较远。(2) 同一类别图像的哈希码应该彼此接近。(3) 最小化量化误差。所提出的方法在三个图像检索数据集上进行评估。结果表明，所得到的哈希中心始终由推导的最小距离分隔，并且所提出的方法优于现有的深度哈希方法。

2 相关工作

以往的图像检索哈希方法通常使用手工设计的视觉特征作为图像表示，然后通过投影或量化生成哈希码。近年来深度哈希 [11] [12] [13] [14] 在生成更有效和紧凑的二进制表示方面越来越受欢迎。深度哈希方法可以大致分为成对方法、三元组方法和逐点方法。

2.1 成对和三元组深度哈希方法

成对方法 [17] [18] [19] 通过使用数据对之间的成对相似性学习哈希函数，而三元组方法 [13] [20] 利用数据三元组之间的相对相似性。然而，成对方法和三元组方法都只能捕捉数据的部分相似关系，可能会导致训练效率低、数据分布覆盖不足和正/负对不平衡问题 [15]。

2.2 逐点深度哈希方法

逐点方法可以进一步分为假设类别标签可用的方法 [21] [22] [23] 和派生哈希中心的方法 [15] [24] [25]，其中同一语义类别的数据点被分配到相同的哈希中心。本文基于数据高效的逐点深度哈希方法，通过保证哈希中心之间的理论最小距离并展示实证性能提升来改进该方法。逐点方法使用数据点的标签作为监督来学习哈希函数，本方法的一个关键特性是任意两个哈希中心之间的距离应足够大，以便能够清晰地将远离的语义类别的数据点分开。DPN [24] 提出了一种优化算法来获得哈希中心。CSQ [15] 使用 Hadamard 矩阵和伯努利采样，使得哈希中心的平均距离为哈希码长度的一半。OrthoHash [7] 采用类似的方式获取哈希中心，并将其用于单一目标。

2.3 中心相似性量化方法

对于哈希中心方法来说，生成高质量的哈希中心尤为重要。最直观的哈希中心生成方法是根据图像和视频数据本身的分布来生成中心，但 CSQ [15] 研究发现，通过数据生成的哈希中心的效果并不如预先定义好的哈希中心。CSQ 假设每个中心与其他中心的距离应该比与之相关的哈希码更远。因此，不同的数据对可以更好地分离，相似的数据对可以内聚，将哈希中心定义为在 K 维汉明空间中，具有平均成对距离满足条件的点集 $\mathcal{C} = \{c_i\}_{i=1}^{m} \subset \{0,1\}^K$。哈希中心有两个生成的方法，具体如下 2.3.1 和 2.3.2 所述。

2.3.1 利用 Hadamard 矩阵生成哈希中心

对于 $K \times K$ 的 hadamard 矩阵 $H_K = \left[h_a^1; \ldots; h_a^K\right]$ 来说，具有以下优秀的性质：1）它是一个方阵，其中行向量 h_a^1 互相正交，即任意两个行向量的内积 $\langle h_a^i, h_a^j \rangle = 0$ 。任意两个行向量之间的汉明距离为 $D_H\left(h_a^i, h_a^j\right) = \frac{1}{2}\left(K - \langle h_a^i, h_a^j \rangle\right) = \frac{K}{2}$ 。因此，可以从这些行向量中选择哈希中心。2）矩阵的大小 K 是 2 的幂（即 $K = 2^n$ ），与哈希码的位数习惯保持一致。3）它是一个二进制矩阵，其元素要么是-1，要么是 +1。因此，可以将所有的-1 替换为 0，以获得在 $\{0,1\}^K$ 中的哈希中心。4）行（列）和的规律：Hadamard 矩阵每一行（或每一列）的和为 0。5）hadamard 矩阵的求解是一个递归结构。为了从哈达玛矩阵中采样哈希中心，首先建立

了一个 $K \times K$ 哈达玛矩阵，如下：

$$H_K = \begin{bmatrix} H_{2^{n-1}} & H_{2^{n-1}} \\ H_{2^{n-1}} & -H_{2^{n-1}} \end{bmatrix} = H_2 \otimes H_{2^{n-1}} \tag{1}$$

$\otimes$ 表示 Hadamard 乘积，$K = 2^n$，初始的两个因素是 $H_1 = [1]$ 和 $H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$。当 $m \leqslant K$ 时，每一行作为哈希中心。当 $K < m \leqslant 2K$ ，使用两个 Hadamard 矩阵的组合 $H_{2K} = [H_K, -H_K]^\top$ 来构建哈希中心。

2.3.2 利用 Bern 分布生成哈希中心

当生成的矩阵并不是方阵，以及当 $m > 2K$ 或 $K \neq 2^n$ 时，则通过 Hadamard 矩阵的方法并不适用。因此，CSQ 提出了第二代方法，通过随机抽样生成每一个哈希中心。具体而言，每个中心 c_i 的每个位都从伯努利分布 $Bern(0.5)$ 中抽样，其中如果 $x \sim \text{Bern}(0.5)$ ，则 $P(x = 0) = 0.5$ ，也就是哈希中心的距离的期望为 $K/2$ ，满足哈希中心的要求。获得一组哈希中心后，将训练数据样本 X 与它们各自对应的中心关联起来，计算中心相似度。

2.3.3 损失函数

损失函数主要由中心相似性度量损失和量化损失两部分构成：

$$\begin{aligned} &\min_{\Theta} L_T = L_C + \lambda_1 L_Q \\ &= \frac{1}{K} \sum_{i}^{N} \sum_{k \in K} \left[c'_{i,k} \log h_{i,k} + \left(1 - c'_{i,k}\right) \log \left(1 - h_{i,k}\right) \right] + \lambda_1 \sum_{i}^{N} \sum_{k=1}^{K} \left(\log \cosh \left(|2h_{i,k} - 1| - 1 \right) \right. \end{aligned} \tag{2}$$

其中，L_C 表示的是中心相似性量化，本质上是通过最大化似然概率来计算哈希码的对数最大后验（MAP）估计，该损失函数是为了保证哈希码接近其对应的哈希中心，并远离其他的哈希中心。L_Q 是衡量哈希码的量化损失, 这里使用双峰拉普拉斯先验进行量化，其定义为

$$L_O = \sum^{N} \left(\| |2h_i - \mathbf{1} \mid -\mathbf{1}| \|_1 \right) \tag{3}$$

，其中 $\mathbf{1} \in \mathbb{R}^K$ 是一个全为 1 的向量。由于是一个非光滑函数，这使得计算其导数变得困难。采用平滑函数 log cosh 来替代它。因此，$|x| \approx \log \cosh x$。

2.3.4 个人总结

本篇论文针对哈希学习过程中出现的，1）衡量全局相似性难度大，2）数据不平衡问题（不相似的数据的数量远远超过相似的数据的数量），提出了直接根据类别生成哈希中心，以哈希中心的相似度来衡量全局的相似度。这样一是可以减轻计算成本，二是可以考虑全局的信息。在信息检索的领域上有独特的优势。此外，针对多标签的数据集，本文提出了针对多标签的数据集，综合样本的多个标签提出质心的概念。解决了多标签哈希中心的问题。这是一个全新的哈希方法，实现了通过少量的哈希中心来实现衡量全局相似性这一目的。

2.3.5 本文优化思路

基于 CSQ 生成的哈希中心可能会降低检索性能的问题，作者提出了一种优化方法，可以保证任意一对哈希中心之间的最小距离。具体而言，采用了编码理论中的 Gilbert-Varshamov 界限，这有助于获得一个较大的最小距离，同时确保我们的优化方法的实证可行性。

3 本文方法

3.1 本文方法概述

图像的哈希学习的目标是找到一个函数 $M : \mathcal{X} \to \{-1, 1\}^q$ ，将图像 $x \in \mathcal{X}$ 转换为长度为 q 的哈希码 $\to \{-1, 1\}^q$ ，使得相似的图像具有相邻的哈希码，而不相似的图像具有相距较远的码。对于深度哈希方法，M 通常由一个输出哈希码的深度网络层实现。

由于 CSQ 在构建的哈希中心之间的过程中，采取了 Hadamard 矩阵和伯努利分布的方法，也就是有可能出现两个哈希中心的距离为 0 的情况。也就是说，最小距离方面缺乏最坏情况的保证，哈希中心之间的距离可以任意接近。为了解决这个问题，本文提出了一个两步的深度哈希框架，阶段一是生成哈希中心，阶段二是设计深度神经网络。如图 1所示，所提出的方法采用两阶段流程。

在第一阶段，作者开发了一个优化过程，以生成一组哈希中心，并在最坏情况下，确保任意两个中心之间的汉明距离不小于最小距离 d。更重要的是使用了 Gilbert-Varshamov 界限来确定一个较大的最小距离 d，同时确保优化过程的可行性。每个哈希中心被分配给一个图像类别，通过哈希中心来衡量全局相似性。

第二阶段，作者使用三个损失函数训练深度哈希网络：一个针对哈希中心的损失函数，以鼓励输入图像的哈希码靠近其类别的哈希中心，但远离其他中心；一个用于相似图像的损失函数，以鼓励同一类别中图像的哈希码相互接近；以及一个用于减少量化误差的损失函数。接下来将分别说明这两个阶段（3.2 和 3.3）。

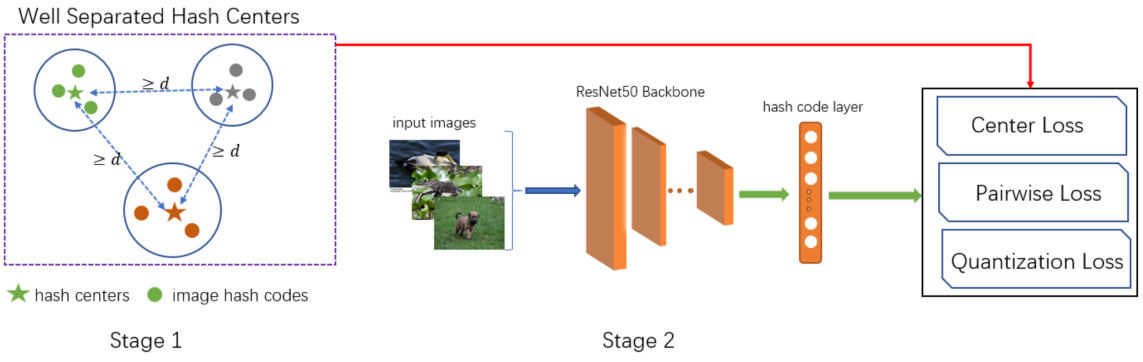

图 1. 深度哈希中心神经网络架构图，提出的方法由一个两阶段的流程组成。

3.2 通过优化生成哈希中心

3.2.1 目标函数设计

第一阶段，设计了生成一组哈希中心的优化方法。每个哈希中心分配给一个图像类别，使用 Gilbert-Varshamov 界确定一个较大的最小距离 d ，对于 c 类图像，作者试图学习 c 个哈希中心 $h_1, h_2, \ldots, h_c$ ，旨在最大化任意两个哈希中心之间的平均距离，并确保任意两个中心之间的汉明距离不小于一个最小距离 d ，优化目标可以表述为：

$$\begin{aligned} &\max_{h_1,\ldots,h_c\in\{-1,1\}^q} \frac{1}{c(c-1)} \sum_i \sum_{j:j\neq i} \|h_i - h_j\|_H \\ &\text{s.t. } \|h_i - h_j\|_H \geq d(1 \leq i, j \leq c, i \neq j) \end{aligned} \tag{4}$$

其中 $\|\cdot\|_H$ 是汉明距离，d 是最小距离参数。让 $\|\cdot\|_2$ 表示 ℓ_2 范数。由于 $h_i, h_j \in \{-1, 1\}^q$，可以得到 $4\|h_i - h_j\|_H = \|h_i - h_j\|_2^2 = h_i^T h_i + h_j^T h_j - 2h_i^T h_j = 2q - 2h_i^T h_j$。$q$ 是哈希码的长度。因此，最大化 $\|h_i - h_j\|_H$ 等同于最小化 $h_i^T h_j$ ，$\|h_i - h_j\|_H \geq d$ 等同于 $h_i^T h_j \leq q - 2d$。因此，优化目标函数可以转化为：

$$\begin{aligned} &\min_{h_1,\ldots,h_c\in\{-1,1\}^q} \sum_i \sum_{j:j\neq i} h_i^T h_j \\ &\text{s.t. } h_i^T h_j \leq q - 2d(1 \leq i, j \leq c, i \neq j) \end{aligned} \tag{5}$$

在一般情况下，由于二进制约束 $h_1, h_2, \ldots, h_c \in \{-1, 1\}^q$，方程（4）中的优化目标是一个 NP 难问题。作者采用一种交替更新的方法，固定其他哈希中心 $h_j(1 \leq j \leq c, j \neq i)$ 求解更新其中一个哈希中心 h_i 。通过固定所有的 $h_j(j \neq i)$，将关于 h_i 的子问题形式化为：

$$\begin{aligned} &\min_{h_i\in\{-1,1\}^q} \sum_{j:j\neq i} h_i^T h_j \\ &\text{s.t. } h_i^T h_j \leq q - 2d(1 \leq j \leq c, j \neq i) \end{aligned} \tag{6}$$

关于如何解开二进制的约束，采用 ℓ_p-box 算法，旨在将二进制约束化为易于求解和表达的约束条件。ℓ_p-box 算法最初的表现形式是：

$$\min_{\mathbf{x}\in\{0,1\}^n} f(\mathbf{x}), \text{ s.t. } \mathbf{x} \in \mathcal{C} \tag{7}$$

关于这个二进制约束，可以把它转化为一个球体和一个盒的交集，通过盒子和球体对原始的二进制约束进行转化，约束转化的过程如下所示：

$$\mathbf{x} \in \{0, 1\}^n \Leftrightarrow \mathbf{x} \in [0, 1]^n \cap \left\{ \mathbf{x} : \left\| \mathbf{x} - \frac{1}{2}\mathbf{1}_n \right\|_p^p = \frac{n}{2^p} \right\} \tag{8}$$

因此，本文中的二进制约束 $v \in \{-1, 1\}^q$ 等同于 $v \in [-1, 1]^q \bigcap \{v : \|v\|_p^p = q\}$ 。因此，公式（7）可以重新表述为以下等价形式。

$$\begin{aligned} &\min_{h_i,v_1,v_2} \sum_{j:j\neq i} h_i^T h_j \\ &\text{s.t. } h_i^T h_j \leq q - 2d(1 \leq j \leq c, j \neq i) \\ &h_i = v_1, h_i = v_2, v_1 \in \mathcal{V}_{box}, v_2 \in \mathcal{V}_{sph} \end{aligned} \tag{9}$$

其中，$\mathcal{V}_{\text{box}} = \{v : -1_q < v < 1_q\}$，$\mathcal{V}_{sph} = \{v : \|v\|_2^2 = q\}$。首先，引入辅助变量 v_3 作为一个简单的范围约束，$v_3 \in R_+^{c-1}$，其中 $H_{\sim i} = [h_1, \ldots, h_{i-1}, h_{i+1}, \ldots, h_c]$ 表示由 $h_j (1 \le j \le c, j \ne i)$ 组成的矩阵，$R_+^{c-1} = \{v : v \in [0, +\infty)^{c-1}\}$。因此，$h_i^T h_j \le q - 2d$ 可以被替换为等式约束 $h_i^T H_{\sim i} + v_3 = (q - 2d)1_{c-1}$ 。方程（8）中的问题可以进一步表示为：

$$\begin{aligned} &\min_{h_i, v_1, v_2, v_3} \sum_{j:j \ne i} h_i^T h_j \\ &\text{s.t. } h_i^T H_{\sim i} + v_3 = (q - 2d)1_{c-1}, v_3 \in R_+^{c-1}, \\ &h_i = v_1, h_i = v_2, v_1 \in \mathcal{V}_{box}, v_2 \in \mathcal{V}_{sph}. \end{aligned} \tag{10}$$

这里使用增广拉格朗日方法，将约束条件加入到目标函数之中联合求解，等式（9）被改写为

$$\begin{aligned} L(h_i, v_1, v_2, v_3, k_1, k_2, k_3) &= \sum_{j \ne i} h_i^T h_j + k_1^T (h_i - v_1) + \frac{\mu}{2} \|h_i - v_1\|_2^2 + k_2^T (h_i - v_2) \\ &+ \frac{\mu}{2} \|h_i - v_2\|_2^2 + k_3^T \left(h_i^T H_{\sim i} + v_3 - e\right) + \frac{\mu}{2} \left\|h_i^T H_{\sim i} + v_3 - e\right\|_2^2 \\ &\text{s.t. } v_1 \in V_{box}, v_2 \in V_{sph}, v_3 \in R_+^{c-1} \end{aligned} \tag{11}$$

其中，$e = (q - 2d)1_{c-1}$，k_1, k_2, k_3 是拉格朗日乘子。

3.2.2 推导最小距离 d

对于公式（4）中的目标，需要设置最小距离参数 d，以便对于任意两个中心 $h_i, h_j (i \ne j)$，汉明距离 $\|h_i - h_j\|_H$ 不小于 d。这里需要一个较大的 d，使得哈希中心彼此之间距离较远。但是 d 不能太大，以确保公式（4）中最小距离约束的可行性。在这里，作者通过采用编码理论中的 Gilbert-Varshamov 界来确定一个较大的 d。具体而言，对于 c 个 q 位二进制码 $h_i \in \{-1, 1\}^q (1 \le i \le c)$ ，任意两个码之间的汉明距离至少为 d ，Gilbert-Varshamov 界定了 c 个 q 位码，使得任意两个码之间的最小汉明距离为 d，只要 c、q 和 d 满足以下条件：

$$\frac{2^q}{c} \le \sum_{i=0}^{d-1} \begin{pmatrix} q \\ i \end{pmatrix} \tag{12}$$

为了生成 c 个 q 位哈希中心，只要设置一个满足方程（11）的 d，Gilbert-Varshamov 界限确保了方程（4）中最小距离约束的可行性。因此，为了获得一个大的 d，只需要找到满足方程（11）的 d 的最大值。函数 $f(d) = \sum_{i=0}^{d} \begin{pmatrix} q \\ i \end{pmatrix}$ 在 d 方面是单调递增的。因此，设 $d^\star$ 是满足方程（11）的 d 的最大值，可以得到：

$$\begin{cases} \frac{2^q}{c} \le \sum_{i=0}^{d^\star - 1} \begin{pmatrix} q \\ i \end{pmatrix} \\ \frac{2^q}{c} > \sum_{i=0}^{d^\star - 2} \begin{pmatrix} q \\ i \end{pmatrix} \end{cases} \tag{13}$$

由于 $d^\star$ 是在 $\{1, 2, \ldots, q\}$ 中的整数，可以通过穷举搜索轻松找到它，即为所需的最小距离 d。

3.3 训练哈希网络

通过获取的哈希中心作为监督信息，作者训练了一个深度哈希网络，可以将输入图像转换为哈希码。如图 1 所示，这个深度网络由三个块组成。第一个块是以 ResNet-50 作为骨干网络，由堆叠的卷积层组成，用于捕捉输入图像的特征表示。第二个块是哈希码层，采用具有 TanH 激活的全连接层实现，将图像特征转换为近似哈希码向量，所有元素都限制在范围 $[-1\ +1]$ 内。在预测时，使用简单的量化将这个近似哈希码转换为二进制码。第三个块由三个损失函数组成。利用从第一阶段获得的哈希中心，第一个损失函数旨在使图像的哈希码接近其类别的哈希中心，但同时远离其他类别的哈希中心。第二个损失函数是一种成对损失，使同一类别中的一对图像具有相邻的哈希码。第三个损失函数是减少量化误差。接下来将分别说明这些损失函数。

3.3.1 哈希中心的损失

获得 c 个哈希中心后，将一个中心分配给 c 个图像类别之一，令图像输出的哈希码接近分配给该图像类的哈希中心，并且远离其他哈希中心。具体而言，给定 c 个哈希中心 $h_1, h_2, \ldots, h_c$，N 个图像 $I_1, I_2, \ldots, I_N$，其输出哈希码分别为 $b_1, b_2, \ldots, b_N$，对于哈希中心的损失函数定义为：

$$L_C = -\frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{c} y_{j,i}\log P_{j,i} + (1-y_{j,i})\log\left(1-P_{j,i}\right) \tag{14}$$

其中，

$$P_{j,i} = \frac{\exp\left[-S\left(b_j, h_i\right)\right]}{\sum_{m=1}^{c}\exp\left[-S\left(b_j, h_m\right)\right]} \tag{15}$$

$S(x,y)$ 为 x 和 y 之间相似性的度量，如果图像 I_j 属于第 i 类的哈希中心为 h_i，则 $y_{j,i}=1$，否则 $y_{j,i}=0$。这里使用缩放余弦相似性作为相似性度量，因此方程中的 $P_{j,i}$ 可以重新表述为：

$$P_{j,i} = \frac{\exp\left[\sqrt{q}\cos\left(b_j, h_i\right)\right]}{\sum_{m=1}^{c}\exp\left[\sqrt{q}\cos\left(b_j, h_m\right)\right]} \tag{16}$$

在这里，将 $\cos(x,y) = \frac{x^T y}{\|x\|_2\|y\|_2}$ 定义为 x 和 y 之间的余弦相似度，q 是哈希码的长度。

3.3.2 相同哈希中心的相似对的损失

考虑图像 I_x 和 I_y，其哈希码分别为 b_x 和 b_y，I_x 和 I_y 属于同一类别，并被分配到哈希中心 h。在公式（15）中，对哈希中心的损失函数使得 b_x 和 b_y 接近哈希中心 h。同时，作者希望属于同一哈希中心的哈希码也尽可能接近，因此设计了一种损失函数，使得相似图像的哈希码的汉明距离变小。该损失函数定义为：

$$L_P = -\sum_{I_x, I_y \text{ are similar}} \log\frac{1}{1+e^{D(b_x,b_y)}} \tag{17}$$

其中 $D(x,y)$ 是 x 和 y 的距离度量。使用负对数似然函数使距离 $D\left(b_x, b_y\right)$ 尽可能小。对于两个二进制码 $h_1, h_2 \in \{-1,1\}^q$，本文采用了缩放的汉明距离 $\frac{1}{q}\left\|h_1-h_2\right\|_H = \frac{q-h_1^T h_2}{2q}$。

由于 b_i 和 b_j 是连续向量，不能用于计算汉明距离。因此，将 $D\left(b_x, b_y\right)=\frac{q-b_i^T b_j}{2q}$ 设置为缩放汉明距离的近似值。因此损失函数可以转化为：

$$L_P=\sum_{I_x, I_y \text{ are similar}} \log\left(1+e^{\frac{q-b_x^T b_y}{2q}}\right) \tag{18}$$

3.3.3 量化损失

由于每个哈希中心都是二进制的，优化难度较大。在提出的网络中，输出的哈希码是连续的，因为哈希码层是通过具有 tanh 激活函数的全连接层实现的。为了减少量化误差，类似于现有的方法 [26]，这里使用具有双峰拉普拉斯先验的损失函数，其定义为：

$$L_Q=\sum_{i=1}^{N}\left\|\left\|b_i\right\|_1-1\right\|_1 \tag{19}$$

其中，$\|\cdot\|_1$ 是 ℓ_1 范数，使用接近双峰的先验可以在训练时对哈希码的生成施加一些约束。具体而言，双峰分布的性质意味着哈希码在生成时更有可能分布在两个不同的区域，而不是聚集在一个单一的区域。这对于哈希码来说是有益的，因为它可以促使哈希码更均匀地分散，而不是偏向于一个特定的方向。在量化损失的上下文中，采用接近双峰的先验可以帮助哈希码更好地适应量化的过程，使得量化后的哈希码在二进制空间内更均匀地分布。这可以减少量化误差，因为通过考虑先验，模型更有可能学得适用于量化任务的哈希码表示。

3.3.4 总体损失函数

将这三个损失函数结合起来，形成了在提出的深度哈希网络中使用的优化目标。

$$L=L_C+\lambda_1 L_P+\lambda_2 L_Q \tag{20}$$

最终，通过一个两步的哈希网络架构不断迭代得到最终结果。

3.3.5 个人总结

本篇论文的重点就是在于，在哈希中心方法的基础上，引入了最小距离约束。表面上来看，最小距离约束的想法很容易想到，但实际上很难得到求解到一个合适的最小距离的方法，并且在这个约束下求解目标函数。而本文就是提出了这样的方法，在自己的实验过程中也可以考虑把 CSQ 类的方法引入到自己的哈希学习框架之中。

4 复现细节

4.1 创新点

（1）由于将实值数据映射成二进制哈希码会带来较大的信息损失，因此我考虑在损失函数之中加入原始特征与哈希码之间的相互重构方法以拟合两者之间的误差，期望学习到的二进制哈希码可以保留更多的信息。

（2）由于在目前的跨模态哈希方法中，需要衡量全局的相似性，而这无疑会带来较大的计算成本。然而本文设计的哈希中心方法，可以通过哈希中心来衡量全局相似性达到较好的效果，因此考虑将哈希中心的方法引入到跨模态哈希之中。

4.1.1 创新一

将复杂的数据映射为 01 代码会损失大量信息，因此我在本文的基础上考虑加入了原始特征和哈希码的重构以拟合两者误差。研究表明，通过矩阵分解可以学习原始数据集的潜在语义特征 [27]。哈希矩阵 B 可以视作原始样本的语义表示。原始特征与哈希矩阵之间的关系可以通过学习矩阵 U 来构建，这使得哈希矩阵 B 能够保留原始数据的语义特征并减少信息损失。因此，可以得到以下方程：

$$L_M = \|X - BU\|_2^2 \tag{21}$$

将该损失函数加入到（19）中的损失函数，因此总体的损失函数可以表示为：

$$L = L_C + \lambda_1 L_P + \lambda_2 L_Q + \lambda_3 L_M \tag{22}$$

4.1.2 创新二

由于跨模态哈希中需要考虑全局相似度，这一过程往往需要消耗极大的计算成本。而哈希中心的方法可以很好地解决这一问题，因此这里以经典的跨模态哈希方法 DCMH [28] 作为基准，将哈希中心的方法引入到跨模态哈希之中。整个 DCMH 模型如图 2所示，它是一个端到端的学习框架，通过无缝集成两个部分：特征学习部分和哈希码学习部分。在学习过程中，每个部分都可以向另一个部分提供反馈。

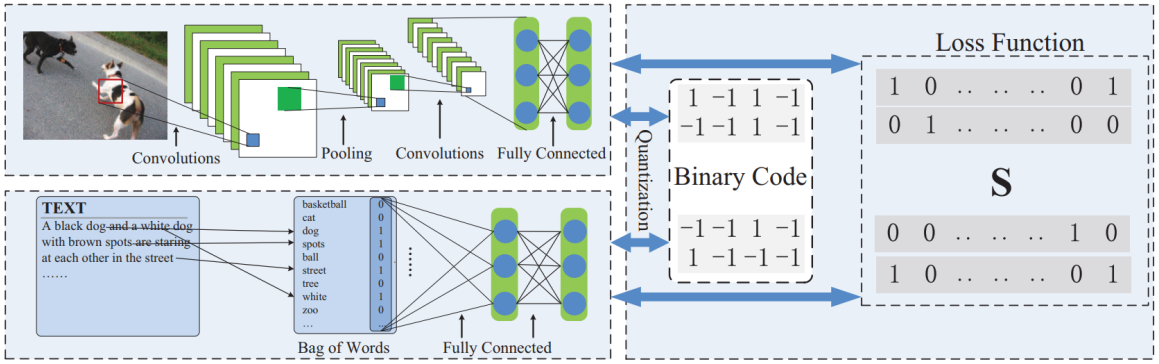

图 2. DCMH 模型网络架构图

简单来说，这里图像特征提取部分采用了 CNN 的网络架构，文本特征提取采用了 MLP 的网络架构。为了引入哈希中心的方法，我将该框架中原本 CNN 的网络架构替换成本文作者所设计的哈希中心的网络架构，将两者方法进行结合，取名为 CCMH，测试其性能效果，模型结构如图 3所示：

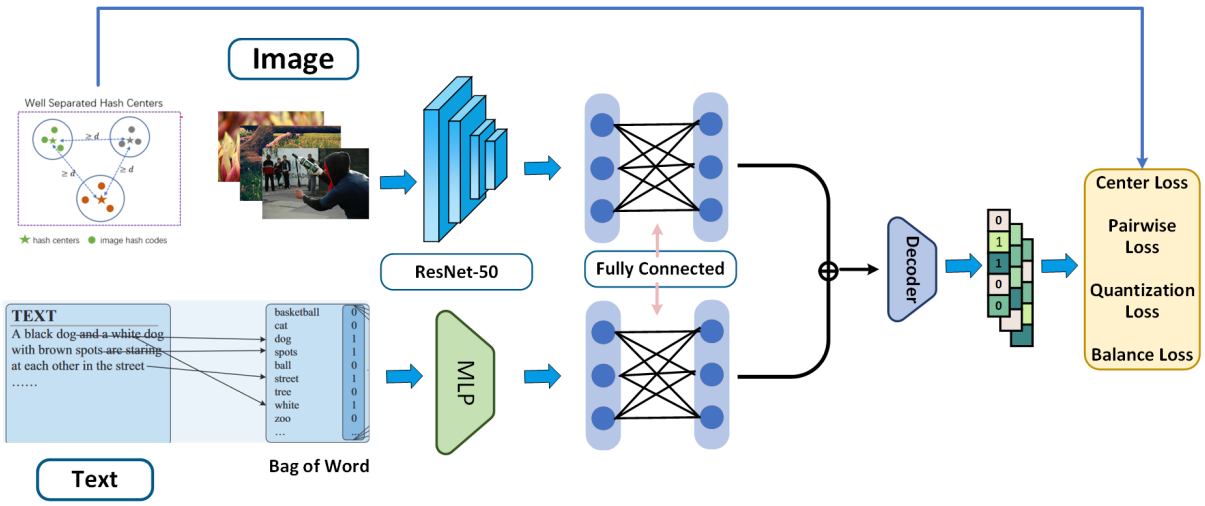

图 3. 我所设计的跨模态哈希中心网络架构

设 $f(\mathbf{x}_i;\theta_x)\in\mathbb{R}^c$ 代表点 i 的学习图像特征，对应于图像模态的通过哈希中心网络输出。此外，设 $g(\mathbf{y}_j;\theta_y)\in\mathbb{R}^c$ 代表点 j 的学习文本特征，对应于文本模态的深度神经网络输出。其中，θ_x 是图像模态的哈希中心网络的网络参数，θ_y 是文本模态的深度神经网络的网络参数。这里我考虑了每个图像样本与哈希中心之间的损失，加上 DCMH 原本设计的损失函数我设计的模型的目标函数定义如下：

$$\begin{aligned}\min_{\mathbf{B}^{(x)},\mathbf{B}^{(y)},\theta_x,\theta_y}\mathcal{J}=&-\sum_{i,j=1}^{n}\left(S_{ij}\Theta_{ij}-\log\left(1+e^{\Theta_{ij}}\right)\right)+\gamma\left(\left\|\mathbf{B}^{(x)}-\mathbf{F}\right\|_F^2+\left\|\mathbf{B}^{(y)}-\mathbf{G}\right\|_{F'}^2\right)\\&+\eta\left(\|\mathbf{F1}\|_F^2+\|\mathbf{G1}\|_F^2\right)-\frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{c}y_{j,i}\log P_{j,i}+(1-y_{j,i})\log\left(1-P_{j,i}\right)\\&\text{s.t. }\mathbf{B}^{(x)}\in\{-1,+1\}^{c\times n},\mathbf{B}^{(y)}\in\{-1,+1\}^{c\times n},\end{aligned}\tag{23}$$

其中，$\mathbf{F}\in\mathbb{R}^{c\times n}$ 由 $\mathbf{F}_{*i}=f(\mathbf{x}_i;\theta_x)$ 所组成；$\mathbf{G}\in\mathbb{R}^{c\times n}$ 由 $\mathbf{G}_{*j}=g(\mathbf{y}_j;\theta_y)$ 组成，$\Theta_{ij}=\frac{1}{2}\mathbf{F}_{*i}^T\mathbf{G}_{*j}$。$\mathbf{B}_{*i}^{(x)}$ 是图像 $\mathbf{x}_i$ 的哈希码，$\mathbf{B}_{*j}^{(y)}$ 是文本 $\mathbf{y}_j$ 的哈希码。公式中的第一项可以保持图像特征表示 F 和文本特征表示 G 的跨模态相似度。通过优化公式中的第二项，可以通过 sgn 符号函数学习哈希码，即 $\mathbf{B}^{(x)}=\text{sign}(\mathbf{F})$ 和 $\mathbf{B}^{(y)}=\text{sign}(\mathbf{G})$。因此，可以将 F 和 G 视为 $\mathbf{B}^{(x)}$ 和 $\mathbf{B}^{(y)}$ 的连续替代。由于 F 和 G 能够保持跨模态相似性，可以期望二进制哈希码 $\mathbf{B}^{(x)}$ 和 $\mathbf{B}^{(y)}$ 也能够保持中的跨模态相似性。公式中的第三项用于使哈希码的每一位在所有训练点上保持平衡，这个约束可以用来最大化每一位提供的信息。第四项为哈希中心的损失，约束每个样本的哈希码趋近于其对应的哈希中心。通过总体的损失函数指导整个跨模态哈希中心网络架构。

4.2 与已有开源代码对比

本篇论文作者在 github 上公布了代码，网址为：https://github.com/Wangld5/Center-Hashing 。在本地环境中，对代码进行了调试修改使得其可以正常运行。跨模态哈希有一个整体的代码库，网址为：https://github.com/WangGodder，经过我修改后使用。并且，基于此方法设计了两个改进方法，具体修改如下所述：

（1）首先调通的原始文件，具体是对一些代码细节进行修改使得其可以正常运行。

（2）在 Center-Hashing-main 文件中加入了一个新的损失函数设计，如（21）（22）所示。

（3）在 CSQ+DCMH 文件夹中，首先我是在跨模态哈希方法的模型架构上进行修改，加入了哈希中心方法，整体架构写于 CCMH.py 中。

4.3 实验环境搭建

运行此代码所需的实验环境如下：

1）Python 3.7；

2）Pytorch 2.1.0；

3）其他（numpy, tqdm, 等）

4.4 代码使用说明

（1）首先需要下载 Stanford Cars 数据集、ImageNet 数据集和 MIRFLICK-25K 数据集存放到 dataset 文件夹中，dataset 文件夹需要与 Center-Hashing-main 文件夹和 CSQ+DCMH 文件夹同时处于一个文件夹下。

源码以及创新一使用说明：

（2）运行 optimizeAccel.py 文件，按照不同数据集不同类别数目生成相应的哈希中心文件，存储在 centerswithoutVar 文件夹中。

（3）运行 main.py 文件夹即可测试创新一的实验结果。

创新二使用说明：

(4)CCMH.py 文件夹为我搭建的跨模态哈希中心网络架构,运行 script 文件夹下的 main.py 文件即可测试创新二的实验结果。

5 实验结果分析

本节对实验所得结果进行分析，详细对实验内容进行说明，实验结果进行描述并分析。

5.1 数据集介绍

Stanford Cars 数据集是一个用于车辆识别的计算机视觉数据集，由斯坦福大学提供。该数据集包含来自 Web 上的超过 16,000 张汽车图片，涵盖了 196 个汽车类别。每个汽车类别都有大约 80 张不同角度和视角的汽车图像。ImageNet 数据集包含 100 个类别的 143,495 张图像，其中使用 10,000 张图像进行训练（每个类别 100 张图像），使用 5,000 张查询图像进行测试，剩下的作为检索数据库。MIRFLICK-25K 数据集包含从 Flickr 网站收集的 25,000 张图片。每张图片都与几个文本标签相关联。因此，每个点都是一个图像-文本对。选择那些至少有 20 个文本标签的点进行实验。每个点的文本表示为一个 1386 维的词袋向量。对于基于手工特征的方法，每个图像都用一个 512 维的 GIST 特征向量表示。此外，每个点都手动注释了 24 个唯一标签之一。

5.2 复现结果分析

首先我对原文进行了代码复现，如表 1和 2所示，在 Stanford Cars 和 ImageNet 数据集上，基本上达到了原论文的实验结果。然而，文中所展示的 NABirds 数据集，我的测试结果有较大的偏差，猜测是由于 NABirds 数据集本身有 555 个类别，因此在生成哈希中心的过程中有可能会使得哈希中心距离过近导致结果不佳。

哈希码长度	原文	我的测试
16bits	0.8579	0.8508
32bits	0.8731	0.8715
64bits	0.8814	0.8733

表 1. Stanford Cars 的结果

哈希码长度	原文	我的测试
16bits	0.8639	0.8586
32bits	0.8863	0.8883
64bits	0.9019	0.8968

表 2. ImageNet 的结果

5.3 创新一结果分析

创新一本节是测试在加入一个相互重构的损失函数设计之后算法性能的改变，由 3和 4所示可以发现，在加入新的损失函数之后，整体算法并没有提升，甚至算法性能略略下降。在检测具体损失函数的值已经观察 loss 的下降过程可以发现，由于实值特征和二进制哈希码之间差距过大，直接加入这一项损失回极大的增加损失值，导致效果并不十分理想。

哈希码长度	原文	我的测试
16bits	0.8579	0.8475
32bits	0.8731	0.8679
64bits	0.8814	0.8748

表 3. Stanford Cars 的结果

哈希码长度	原文	我的测试
16bits	0.8639	0.8571
32bits	0.8863	0.8864
64bits	0.9019	0.8850

表 4. ImageNet 的结果

5.4 创新二结果分析

I 表示图像，T 表示文本，而 $I \to T$ 则表示使用图像数据查询文本数据；$T \to I$ 表示使用文本数据查询图像数据。如表 5和 6，展示了我设计的第二个算法在 MIRFLICK-25K 数据集上测试的结果。我将哈希码长度设置为 16、32、64，以测试算法在不同码长下的性能。其余对比算法的测试结果为 DCMH 论文中所截取。可以发现，在引入哈希中心的方法之后，$I \to T$ 的性能相较于基准方法 DCMH 以及许多经典的跨模态哈希方法来说有较大的提升。分析原因在于哈希中心的方法大大增强了在图像特征提取的过程中所提取的图像特征的质量，因此性能有较大幅度的提升。然而，针对 $T \to I$ 的方法则没有较大的提升，而是达到了与基准方法 DCMH 差不多的性能，因为在文本特征提取的过程中仅仅是采用了简单提取方法。总结而言，总体效果仍然良好，足以证明该创新方法具有一定的优势。

算法	16bits	32bits	64bits
DCMH	0.7410	0.7465	0.7485
SePH	0.6573	0.6603	0.6616
STMH	0.5921	0.5950	0.5980
SCM	0.6290	0.6404	0.6480
CMFH	0.5818	0.5808	0.5805
CCA	0.5695	0.5663	0.5641
Ours	**0.8022**	**0.8057**	**0.7965**

表 5. $I \rightarrow T$ 的实验结果

算法	16bits	32bits	64bits
DCMH	0.7827	0.7900	0.7932
SePH	0.6480	0.6521	0.6545
STMH	0.5802	0.5846	0.5855
SCM	0.6195	0.6302	0.6366
CMFH	0.5787	0.5774	0.5784
CCA	0.5690	0.5659	0.5639
Ours	**0.7913**	**0.7899**	**0.7924**

表 6. $T \rightarrow I$ 的实验结果

6 总结与展望

总体而言，哈希中心类型的深度哈希方法在近年来展现出了极大的优势，我也针对该方法做出了两点改进工作，主要是希望将哈希中心方法和跨模态任务结合在一起，但目前只尝试了在图像处理过程中应用哈希中心，未来会进一步尝试在图像和文本进行对齐之后，再通过哈希中心的方法对融合对齐后的特征进行学习，猜测此方法可以得到一个较好的结果，待未来进一步尝试。

参考文献

[1] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262, 2004.

[2] Omid Jafari, Preeti Maurya, Parth Nagarkar, Khandker Mushfiqul Islam, and Chidambaram Crushev. A survey on locality sensitive hashing algorithms and their applications. *arXiv preprint arXiv:2102.08942*, 2021.

[3] Huawen Liu, Wenhua Zhou, Hong Zhang, Gang Li, Shichao Zhang, and Xuelong Li. Bit reduction for locality-sensitive hashing. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[4] Brian Kulis and Kristen Grauman. Kernelized locality-sensitive hashing for scalable image search. In *2009 IEEE 12th international conference on computer vision*, pages 2130–2137. IEEE, 2009.

[5] Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. *Advances in neural information processing systems*, 21, 2008.

[6] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2916–2929, 2012.

[7] Wei Liu, Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. Hashing with graphs. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 1–8, Madison, WI, USA, 2011. Omnipress.

[8] Mohammad Norouzi and David J Fleet. Minimal loss hashing for compact binary codes. *mij*, 1:2, 2011.

[9] Wei Liu, Jun Wang, Rongrong Ji, Yu-Gang Jiang, and Shih-Fu Chang. Supervised hashing with kernels. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2074–2081. IEEE, 2012.

[10] Fumin Shen, Chunhua Shen, Wei Liu, and Heng Tao Shen. Supervised discrete hashing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 37–45, 2015.

[11] Yue Cao, Mingsheng Long, Bin Liu, and Jianmin Wang. Deep cauchy hashing for hamming space retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1229–1237, 2018.

[12] Lu Jin, Zechao Li, and Jinhui Tang. Deep semantic multimodal hashing network for scalable image-text and video-text retrievals. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[13] Xiaofang Wang, Yi Shi, and Kris M Kitani. Deep supervised hashing with triplet labels. In *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part I 13*, pages 70–84. Springer, 2017.

[14] Rongkai Xia, Yan Pan, Hanjiang Lai, Cong Liu, and Shuicheng Yan. Supervised hashing for image retrieval via image representation learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28, 2014.

[15] Li Yuan, Tao Wang, Xiaopeng Zhang, Francis EH Tay, Zequn Jie, Wei Liu, and Jiashi Feng. Central similarity quantization for efficient image and video retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3083–3092, 2020.

[16] Baoyuan Wu and Bernard Ghanem. $\ell p - box$ admm: A versatile framework for integer programming. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1695–1708, 2018.

[17] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Philip S Yu. Hashnet: Deep learning to hash by continuation. In *Proceedings of the IEEE international conference on computer vision*, pages 5608–5617, 2017.

[18] Wu-Jun Li, Sheng Wang, and Wang-Cheng Kang. Feature learning based deep supervised hashing with pairwise labels. *arXiv preprint arXiv:1511.03855*, 2015.

[19] Haomiao Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. Deep supervised hashing for fast image retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2064–2072, 2016.

[20] Hanjiang Lai, Yan Pan, Ye Liu, and Shuicheng Yan. Simultaneous feature learning and hash coding with deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3270–3278, 2015.

[21] Shupeng Su, Chao Zhang, Kai Han, and Yonghong Tian. Greedy hash: Towards fast optimization for accurate hash coding in cnn. *Advances in neural information processing systems*, 31, 2018.

[22] Kevin Lin, Huei-Fang Yang, Jen-Hao Hsiao, and Chu-Song Chen. Deep learning of binary hash codes for fast image retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 27–35, 2015.

[23] Huei-Fang Yang, Kevin Lin, and Chu-Song Chen. Supervised learning of semantics-preserving hash via deep convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 40(2):437–451, 2017.

[24] Lixin Fan, Kam Woh Ng, Ce Ju, Tianyu Zhang, and Chee Seng Chan. Deep polarized network for supervised learning of accurate binary hashing codes. In *IJCAI*, pages 825–831, 2020.

[25] Li Yuan, Tao Wang, Xiaopeng Zhang, Francis EH Tay, Zequn Jie, Wei Liu, and Jiashi Feng. Central similarity quantization for efficient image and video retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3083–3092, 2020.

[26] Han Zhu, Mingsheng Long, Jianmin Wang, and Yue Cao. Deep hashing network for efficient similarity retrieval. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 30, 2016.

[27] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[28] Qing-Yuan Jiang and Wu-Jun Li. Deep cross-modal hashing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3232–3240, 2017.