计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2023,32(4):42−51 [doi: 10.15888/j.cnki.csa.009050]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于串行编码校验的深度哈希图像检索①

丁美荣[1], 卢志毅[1,2], 陈殷齐[2]

[1](华南师范大学 软件学院, 佛山 528225)
[2](季华实验室 新型显示技术与装备研究中心, 佛山 528200)
通信作者: 卢志毅, E-mail: 2020023859@m.scnu.edu.cn

摘　要: 现有基于深度学习的哈希图像检索方法通常使用全连接作为哈希编码层, 并行输出每一位哈希编码, 这种方法将哈希编码都视为图像的信息编码, 忽略了编码过程中哈希码各个比特位之间的关联性与整段编码的冗余性, 导致网络编码性能受限. 因此, 本文基于编码校验的原理, 提出了串行哈希编码的深度哈希方法——串行哈希编码网络 (serial hashing network, SHNet). 与传统的哈希编码方法不同, SHNet 将哈希编码网络层结构设计为串行方式, 在生成哈希码过程中对串行生成的前部分哈希编码进行校验, 从而充分利用编码的关联性与冗余性生成信息量更为丰富、更加紧凑、判别力更强的哈希码. 采用 mAP 作为检索性能评价标准, 将本文所提方法与目前主流哈希方法进行比较, 实验结果表明本文在不同哈希编码长度下的 mAP 值在 3 个数据集 CIFAR-10、ImageNet、NUS-WIDE 上都优于目前主流深度哈希算法, 证明了其有效性.

关键词: 深度学习; 图像检索; 编码校验; 串行编码; 哈希学习; 卷积神经网络

引用格式: 丁美荣,卢志毅,陈殷齐.基于串行编码校验的深度哈希图像检索.计算机系统应用,2023,32(4):42–51. http://www.c-s-a.org.cn/1003-3254/9050.html

Deep Hashing Image Retrieval Based on Serial Code Check

DING Mei-Rong[1], LU Zhi-Yi[1,2], CHEN Yin-Qi[2]

[1](School of Software, South China Normal University, Foshan 528225, China)
[2](New Display Technology and Equipment Center, Ji Hua Laboratory, Foshan 528200, China)

Abstract: Existing deep learning-based hashing methods for image retrieval usually cascade several fully connected layers as the hash coding layer and output each bit of the hash code in parallel. This approach treats hash encoding as the information encoding of images and ignores the relevance between bits of the hash code in the coding process and the redundancy of coding, which leads to the limited encoding performance of networks. In light of the principle of code check, this study proposes SHNet, a deep hashing method based on serial encoding. Different from the traditional hashing method, SHNet designs the hash coding network layer structure as a serial mode and verifies the first part of the serial hash codes in the process of generating hash codes, so as to make full use of the relevance and redundancy of codes to generate more informative, more compact, and more discriminative hash codes. Using mAP as the evaluation standard of retrieval performance, the study compares the proposed method with current mainstream hashing methods. The results show that the mAP values of the proposed method under different hash coding lengths are superior to those of the current mainstream deep hashing algorithm on the three datasets of CIFAR-10, ImageNet, and NUS-wide, which proves its effectiveness.

Key words: deep learning; image retrieval; serial coding; hash learning; code check; convolutional neural network (CNN)

① 基金项目: 广东省普通高校人工智能重点领域专项 (2019KZDZX1033); 广东省基础与应用基础研究基金 (2021A1515011171); 广州市基础研究计划基础与应用基础研究项目 (202102080282)
收稿时间: 2022-09-16; 修改时间: 2022-10-21; 采用时间: 2022-11-04; csa 在线出版时间: 2023-01-06
CNKI 网络首发时间: 2023-01-06

1 前言

近些年来, 随着互联网与多媒体技术的高速发展, 每分每秒都有庞大的图像数据在互联网上传输, 如何从大规模的图像数据中快捷、准确地查询检索到用户所需图像数据, 成为图像检索技术领域研究的一个热点问题[1]. 图像检索主要可分为两大类: 以人工标注图像文本信息为主的基于文本的图像检索 (text-based image retrieval, TBIR), 以及以图像语义内容为主的基于内容的图像检索 (content-based image retrieval, CBIR). 其中, 基于内容的图像检索技术是一种相对比较有优势方案, 它是从图像中提取图像的特征信息, 将特征信息存储, 并以特征信息为依据进行图像检索, 不需要繁杂的人工交互和干预. 但一张图像所蕴含的信息, 往往需要高维度特征矢量才能有效表达, 而处理高维度特征矢量需要消耗大量的存储空间和计算资源. 为了保证图像检索质量和计算效率, 哈希方法逐渐受到欢迎, 它能将图像的高维特征信息映射到低维汉明空间的二进制编码 (即哈希编码), 解决上述特征存储空间大、计算复杂度高的难题.

关于哈希图像检索研究, Gionis 等人提出局部敏感哈希 (locality sensitivity hashing, LSH) 算法[2], 首次将哈希应用到图像检索领域, 设计了一种位置敏感的哈希函数随机映射哈希码; Shen 等人提出监督离散哈希 (supervised discrete hashing, SDH) 算法[3], 采用一种离散循环坐标下降法生成哈希编码; Liu 等人提出核监督哈 (kernel-based supervised hashing, KSH) 算法[4], 利用核函数为原始数据提供非线性映射以生成有判别性的哈希码; 迭代量 (iterative quantization) 算法[5] 与乘积量化 (product quantization, PQ) 算法[6] 通过最小化重构误差建立哈希方程; 谱哈希 (spectral hashing, SH) 算法[7] 与锚点图哈希 (anchor graph hashing, AGH) 算法[8] 利用图学习构建数据的哈希编码[9]. 但由于图像数据维度高, 特征提取困难的问题, 上述传统方法都无法达到令人满意的效果.

随着深度学习在计算机视觉领域的成功应用, 越来越多研究表明深度神经网络在图像特征表示方面有出色的性能. 因此, 深度学习被运用到图像哈希检索领域以解决高维度、大规模的图像检索. 深度哈希方法通过卷积神经网络出色的特征表达能力学习哈希方程, 解决传统方法在高维度图像上特征提取难的问题, 生成判别力强的哈希码. Xia 等人提出卷积神经网络哈希 (convolutional neural network hashing, CNNH) 算法[10], 采用两阶段策略, 其中第 1 阶段用非深度学习的方法学习各个数据的近似哈希码, 第 2 阶段通过深度网络学习一个哈希函数来拟合第 1 阶段的哈希码. 深度神经网络哈希 (deep neural network hashing, DNNH) 算法[11] 通过改进 CNNH 算法, 使特征提取和哈希编码可以在同一学习过程中优化. 深度哈希网络 (deep hashing network, DHN)[12] 在 DNNH 算法上进一步改进, 在保留成对相似度的同时, 引入正则化控制量化损失. Liu 等人提出深度监督哈希 (deep supervised hashing, DSH) 算法[13], 引入正则化约束, 使神经网络的实数输出更趋近于离散值. Li 等人提出深度二元组监督哈希 (deep pairwise-supervised hashing, DPSH) 算法[14], 该算法利用标签对, 同时学习图像特征及其哈希编码. 受到 DPSH 算法启发, Wang 等人提出深度三元标签监督哈希 (deep supervised hashing with triplet labels, DTSH) 算法[15], 使用三元组损失函数进行端到端学习. Cao 等人提出哈希网络 (hashnet: deep learning to hash by continuation)[16], 通过平衡训练数据对和引入量化函数的近似来改进 DHN 算法. Li 等人提出深度离散哈希 (deep supervised discrete hashing, DSDH) 算法[17], 将神经网络最后一层的输出直接限制为二进制编码, 并在训练过程中使用交替优化的方式. 深度柯西哈希 (deep cauchy hashing, DCH) 算法[18] 与汉明距离间隔最大化哈西 (maximum-margin hamming hashing, MMHH) 算法[19] 将数据相似对和不相似对建模为一个分布, 利用分布的特性降低由数据不平衡造成的局部优化, 提高对噪声数据的鲁棒性. Jiang 等人提出非对称深度监督哈希 (asymmetric deep supervised hashing, ADSH) 算法[20], 以非对称方式处理查询和数据库, 实现高效训练. Zhang 等人提出改进的深度哈希网络 (improved deep hashing network, IDHN)[21], 在归一化语义标签上将数据对的相似性量化, 通过量化的相似度信息增强多标签数据集的图像检索性能. 中心相似度量化 (Central similarity quantization, CSQ) 算法[22] 与正交哈希 (deep hashing with a single cosine similarity based learning objective, OrthoHash) 算法[23] 使用 Hadamard 矩阵预先设定哈希中心, 通过拉近相似数据与其对应的哈希中心之间的汉明距离进行相似度学习.

以上深度哈希方法都着眼于如何优化相似度度量方法以及降低量化误差, 所设计网络模型的目标本质上

还是分类. 主要有两个原因导致模型性能受限: (1) 16 位哈希编码可以区分百万种类别, 因此必定存在编码冗余 (即哈希编码的冗余性), 而现有方法的网络任务没有利用冗余的编码; (2) 网络哈希编码层仅使用全连接层学习哈希函数, 并行地输出整段哈希编码, 这种方法预先假定了哈希编码每一位的生成相互独立, 忽略了哈希编码自身的内在联系 (即哈希编码的关联性).

因此, 本文首次将深度哈希视为信息论中的编码过程, 利用编码的冗余性和关联性对哈希码进行校验. 在编码传输过程中, 编码信息由于信道的各种干扰而产生错误, 利用冗余编码位可对传输后的内容进行校验. 如图 1 通用编码校验框架所示, 编码校验即由某种判错纠错的目的计算生成校验码, 例如奇偶校验码、汉明编码等. 而由于哈希图像检索生成的编码存在冗余, 因此可将冗余的编码位用作编码校验, 也就是在相同位数编码的情况下, 利用冗余编码位进行编码校验实现更加准确的信息编码. 同时, 由于校验码的生成依据是前部分的信息码, 信息码和校验码拼接组成哈希编码的编码方式增强了哈希编码自身的关联性.

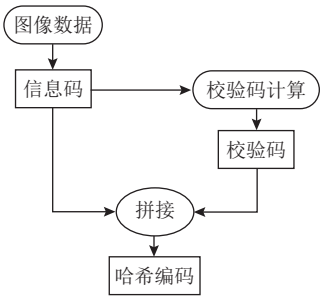


图 1 通用编码校验框架

更具体地, 本文提出的方法将哈希编码分段进行, 在图 1 通用编码校验框架中体现如下: 网络生成的编码 (信息码) 会产生由实数向量转化为二进制编码的量化损失 (即编码在信道传输产生错误的过程), 进而造成信息码错误, 或者生成判别力不足的哈希编码. 此时, 利用冗余的编码位 (校验码) 对存在量化损失的实数向量进行校验, 由于哈希图像检索的目的是对不同类别图像生成具有判别力的哈希编码, 冗余编码位会在含有歧义、判别能力不足的样本上通过校验码计算模块生成额外的、具有判别力的校验码, 因此可以将该校验码和原编码拼接作为哈希编码. 如图 2 编码校验示例所示, 原始的一对在实数空间上具有较大差异的向量在二值化编码之后得到了相同的一对哈希编码, 在汉明空间上不具有判别能力. 此时, 若对原始实数向量添加冗余编码位, 生成额外的校验码, 拼接的实数向量对二值化后, 由于附加的校验码部分而具有判别性, 形成更加可靠的哈希编码.

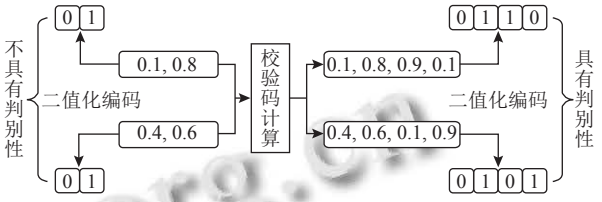


图 2 编码校验示例

而上述过程中, 将原本无关联性的哈希编码分段输出, 在足够信息码位的前提下, 其他冗余的编码位为校验码, 由上一段信息码的特征得到, 最后将各码段拼接作为哈希编码. 本文将这种串行哈希编码的网络命名为 SHNet (serial hash network), 采用 mAP 作为检索性能评价标准, 使用 SHNet 在 4 个公开的数据集 CIFAR-10[24]、ImageNet[25]、MS COCO[26]、NUS-WIDE[27] 上与目前具有代表性的深度哈希方法进行对比实验, 实验结果表明, SHNet 取得了更高的检索性能.

2 基于串行哈希编码的网络

SHNet 检索的详细流程如图 3 所示, 可分为 3 个部分, 第 1 部分介绍本文提出的串行编码方式 (第 2.1 节), 使用图 3 所示串行哈希编码层在每个子编码器输出等长的子编码段, 最后将若干段子编码段拼接作为串行哈希编码层的输出, 即哈希编码; 第 2 部分介绍如何根据数据集的标签生成哈希中心 (第 2.2 节) 预先生成哈希中心; 第 3 部分介绍损失函数 (第 2.3 节), 通过二元交叉熵损失训练缩小图像哈希编码与其对应的哈希中心之间在汉明空间上的距离. 在检索过程, 使用训练好的网络结构作为哈希方程, 将检索图像以及数据库图像编码成二值哈希码, 使用检索图像哈希码与数据库图像的哈希码计算汉明距离, 根据汉明距离排序的方法得到最后的检索结果.

2.1 串行哈希编码网络结构

如图 3 所示, 整体的网络结构由主干网络和串行哈希编码层两部分组成. 主干网络采用预训练的 ResNet50[28] 对图像数据进行特征提取. 哈希编码层部

分如图 3 中串行哈希编码层部分所示, 图中串行哈希编码层采用 4 个子编码器$\mathcal{H}_1$、$\mathcal{H}_2$、$\mathcal{H}_3$、$\mathcal{H}_4$组成. 每个子编码器由编码层和信息层两部分组成 (最后一个子编码器不需要将信息传递给下一个子编码器, 故将其信息层省略). 其中子编码器的编码层负责生成子码段, 子编码器的编码层的信息层负责整合自身的编码信息, 传递给下一个级联的子编码器子编码的详细设计如图 4 所示, 子编码器的编码层部分由两层全连接层 (FC) 组成, 其中第 1 层全连接层后使用 ReLU 作为激活函数, 该层的输出作为编码信息传递给第 2 层全连接层, 将编码信息转换为哈希编码, 在第 2 层全连接层后接一层 batch normalization 层[29], 可以保证编码平衡 (code balance). 在编码层的第一层全连接层后接一个信息层将该子编码器的编码信息传递给下一个子编码器, 信息层由一层全连接层 (FC) 组成, 使用 ReLU 作为激活函数.

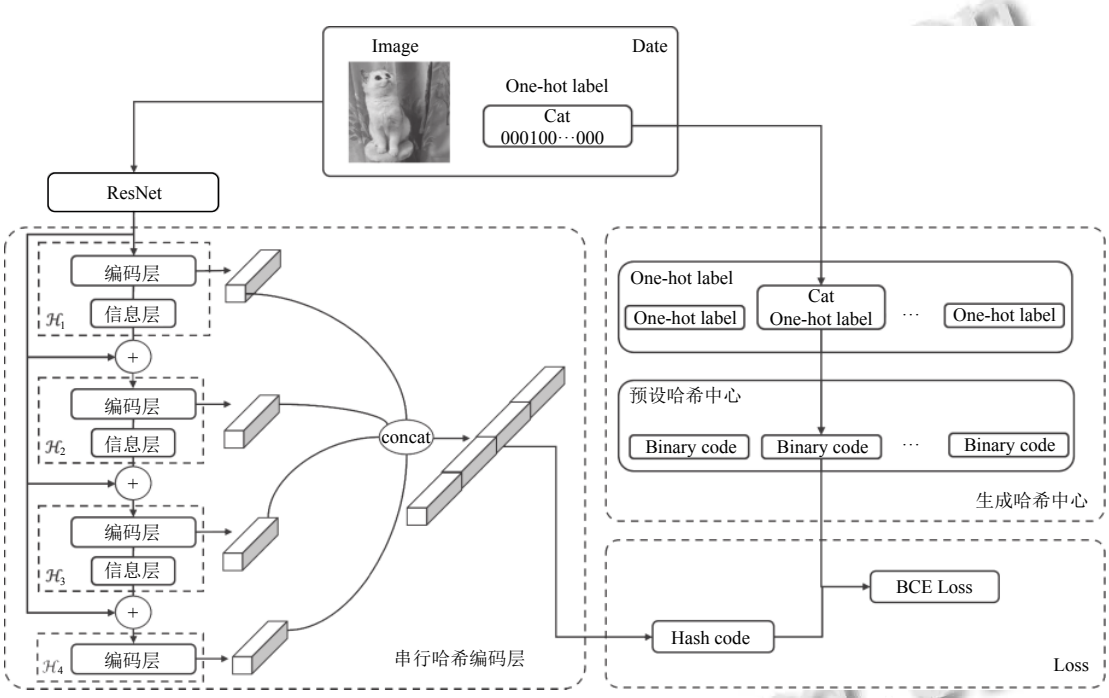


图 3 SHNet 图像检索的工作流程

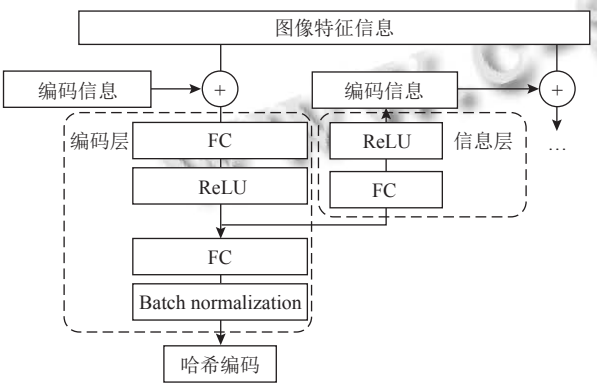


图 4 子编码器的网络结构

设主干网络输出的图像特征向量为x, $\mathcal{H}_n$为第n个子编码器, 将其编码层设为函数ψ_C^n, 信息层设为函数ψ_I^n, 整个串行编码网络层的编码过程如下.

对于第 1 个子编码器, 有:

$$I_1 = \psi_I^1(x) \tag{1}$$

$$h_1 = \psi_C^1(I_1) \tag{2}$$

其中, I_1为第 1 个子编码器的编码信息, h_1为第 1 个子编码器输出的子编码段.

对于第 2 个子编码器:

$$I_2 = \psi_I^2(x + I_1) \tag{3}$$

$$h_2 = \psi_C^2(I_2) \tag{4}$$

由式 (3)、式 (4) 递推可得:

$$I_n = \psi_I^n(x + I_{n-1}) \tag{5}$$

$$h_n = \psi_C^n(I_n) \tag{6}$$

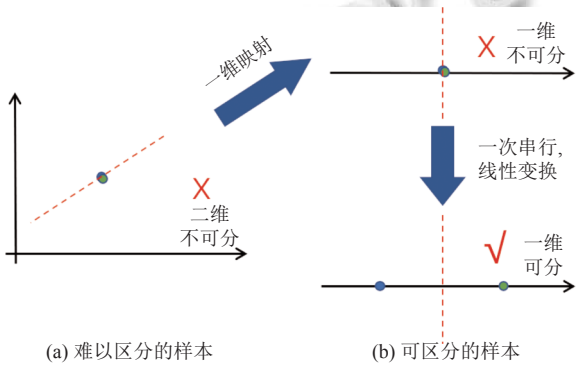
子编码器$\mathcal{H}_1, \mathcal{H}_2, \cdots, \mathcal{H}_n$以串行方式依次输出子编

码段$h_1,h_2,\cdots,h_n$, 最后将编码段拼接生成哈希编码:

$$h=\{h_1,h_2,\cdots,h_n\} \tag{7}$$

除了第 1 个子编码器, 每个子编码器都会收到已输出的子码段的编码信息和主干网络的原始图像特征信息用作编码校验, 输出等长的子编码段. 本文不对每一子编码段功能 (信息码或校验码) 做具体定义, 而是由神经网络为达到更好检索的目的自行训练. 其中, 每个子编码器输出等长的哈希码. 本文所使用的子编码段为 16 bit (过小使编码器深度过大, 有梯度消失或者过拟合的风险, 故此处选择存储单位一字节为一段子编码长度), 如图 3 网络使用 4 个子编码器串行级联, 则最终生成 64 (16×4) bits 的哈希码. 同理, 若最终需要生成 32 (16×2) bits 的哈希码, 则使用 2 个子编码器串行级联.

该串行哈希编码的方式实际上是将全连接的分界面转换为线性空间的嵌套. 如图 5 为一个简单样本分类示例, 假设有两个在二维空间中难以线性区分的样本 (图 5(a) 所示), 那么其在一维空间的映射也同样难以区分, 无论如何优化, 都无法优化出一个能够区分的线性分界面. 然而, 倘若其在一维空间中再做一次线性变换 (全连接层等效于对原始空间做线性变换), 样本在空间中的判别性就能显示出来 (图 5(b) 所示), 这也是利用冗余编码位在分界面上的真正作用. 并行输出哈希编码对应高维度空间的线性分类, 串行输出哈希编码对应低维度空间的线性分类嵌套, 通过将高维度线性分界面分解为多个低维度线性空间的嵌套实现更复杂的分界面, 从而充分利用维度冗余性与比特位关联性来提高模型性能的上限.



(a) 难以区分的样本 (b) 可区分的样本

图 5 样本分类示例

2.2 哈希中心的生成

设训练集包含N个数据$X=\{\{x_i\}_{i=1}^N\}$, 其中$x_i\in\mathbb{R}^D$为需要被哈希编码的数据, 深度哈希方法是学习一个非线性的哈希方程$f:x\mapsto h\in\{0,1\}^K$将输入的数据从$\mathbb{R}^D$映射到K位的汉明空间$\{0,1\}^K$, 输出哈希编码h, 并保证该哈希方程生成的哈希编码h在汉明空间上保留了原始数据x_i在$\mathbb{R}^D$空间上的相似度信息. 生成的哈希中心是相似数据在汉明空间上的聚类中心, 即当相似数据通过哈希方程映射到汉明空间后, 会被拟合到同一个哈希中心上. 哈希中心是数据所属类的抽象, 不同的哈希中心代表不同类数据在汉明空间上的聚类中心, 所以每个哈希中心之间的距离即数据的类间距离应当越大越好. 设K位的汉明空间$H_K\in\{0,1\}^K$, 给定两个任意的二值向量b_i, b_j, 这两个向量的每一位的值为+1 或−1 的概率相等, 均为 0.5. 设向量b_i, b_j的每一位为+1 的概率为p, 则两个向量间的汉明距离的期望值为$\mathbb{E}\left[D_H\left(b_i,b_j\right)\right]=2\cdot K\cdot p(1-p)$, 当$p=0.5$时, $\mathbb{E}\left[D_H\left(b_i,b_j\right)\right]$有最大值, 为$K/2$. 根据汉明距离公式$D_H\left(h_i,h_j\right)=\left(K-\left\langle h_i,h_j\right\rangle\right)/2$可知, 当$\left\langle b_i,b_j\right\rangle=0$, 即$b_i$, b_j为正交向量时, 其汉明距离等于$K/2$, 为K位汉明空间上的汉明距离最大期望值. 所以, 在生成哈希中心矩阵时应尽量使矩阵中任意两个哈希中心c_i, c_j正交, 满足$\mathbb{E}\left[D_H\left(c_i,c_j\right)\right]=K/2$.

Hadamard 矩阵是由+1 和−1 组成的正交矩阵, 即该矩阵的任意两个行向量或列向量都是正交的, 汉明距离为$K/2$, 恰好满足上述需求, 因此可以选取 Hadamard 矩阵的行向量作为哈希中心.

对于单标签数据集, 设数据集 label 为$\mathcal{L}=\{l_1,l_2,\cdots,l_q\}$, q为数据集类的个数, 生成一一对应个数的哈希中心矩阵$C=\left\{c_1,c_2,\cdots,c_q\right\}$, 设生成哈希中心的个数为$q$, 生成哈希编码的位数为$K$, 即汉明空间的维度为$K$, 生成哈希中心的方式如下.

1) 生成大小为 $K\times K$ 的 Hadamard 矩阵$H_K=\left[h_k^i\right]$, 同时生成$H_{2K}=[H_K,-H_K]^{\top}=\left[h_{2k}^i\right]$.

2) 根据编码长度K使用不同的方式生成哈希中心矩阵.

a) 如果哈希码位数$q<K$且$K=2^n$, 哈希中心$c_i=h_k^i$.

b) 如果$K<q<2K$且$K=2^n$, 哈希中心$c_i=h_{2k}^i$.

c) 如果$2K<q$或$K\neq 2^n$, 使用伯努利分布设置哈希中心c的每一位的值x为+1 或−1, 两者概率相等, 即$P(x=-1)=P(x=1)=0.5$, $x\sim Bern(0.5)$. 哈希中心矩阵C中任意两个哈希中心c_i, c_j满足$\mathbb{E}\left[D_H\left(c_i,c_j\right)\right]=K/2$.

3) 将哈希中心矩阵C中所有为–1 的值替换成 0.

4) 得到单标签数据集的哈希中心矩阵$C=\{c_1,c_2,\cdots,c_q\}\subset\{0,1\}^K$, 由于生成哈希中心与数据集的语义标签一一对应, 所以根据数据集中每个数据的标签可得到其对应的哈希中心$C'=\{c'_1,c'_2,\cdots,c'_N\}$, 其中$c'_i$是数据$x_i$的哈希中心.

对于多标签数据集, 由于每个数据x_i同时拥有多个语义标签, 所以不能直接使用与语义标签一一对应的哈希中心作为数据的哈希中心. 首先, 按照单标签的哈希中心生成方法, 使用数据集的语义标签$\mathcal{L}=\{l_1,l_2,\cdots,l_q\}$生成一一对应的二值哈希中心$C=\{c_1,c_2,\cdots,c_q\}$, 其次, 多标签数据集中的数据拥有多个语义标签, 对应了多个哈希中心, 计算这些哈希中心在汉明空间上的中心, 作为该数据的哈希中心. 如图 6 所示, 数据$x_i\in X$包含 3 个语义标签l_a, l_b, l_c, 标签所对应的哈希中心为c_a, c_b, c_c, 设c'_i为c_a, c_b, c_c的中心, 即数据x_i的哈希中心. 统计 3 个哈希中心c_a, c_b, c_c在相同位上的 1 和 0 的数量, 若 1 比 0 的数量大, 则所求c'_i对应位上的值为 1, 反之, 则为 0. 若 0 的数量和 1 的数量相等, 则对应位上取值为 0 或 1, 两者概率均为 50%.

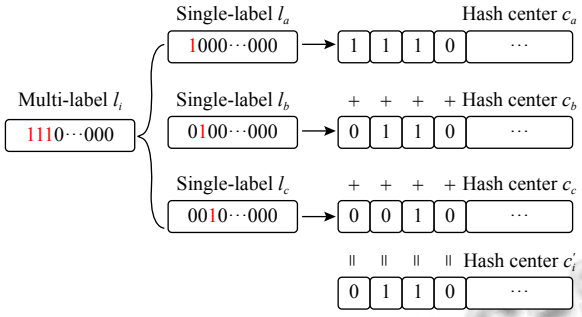


图 6 多标签的哈希中心的生成

综上, 数据X中的每一个数据x_i可以生成其对应的哈希中心c'_i, 得到$C'=\{c'_1,c'_2,\cdots,c'_N\}$. 由此, 有了图像数据的汉明码和图像数据所对应的哈希中心, 可以通过设计损失函数度量图像与其哈希中心的汉明距离, 使网络进行哈希学习, 第 2.3 节将详细介绍损失函数的设计.

2.3 损失函数

通过设计损失函数确定相似度度量策略, 得到神经网络的输入与预期结果的差距, 从而指导网络的训练向正确方向进行. 主流的相似度度量策略有以下几类: 二元组方法, 三元组方法, 中心点方法. 其中二元组方法的算法复杂度为$O(n^2)$, 三元组方法的算法复杂度为$O(n^3)$, 而中心点方法的算法复杂度为$O(n)$, 是 3 种方法中最优的方法, 故本文使用中心点方法设计深度哈希损失函数, 该方法的目标是: 在汉明空间上, 将相似数据拟合至其所属的哈希中心, 减少数据x_i的哈希码h_i与其所属哈希中心c'_i之间的汉明距离$D_H(x_i,c'_i)$. 由于哈希中心为一段二值编码, 其每一位值为 1 或 0, 因此可以把预测哈希编码每一位值当作二分类问题来处理, 使用二元交叉熵 (BCE) 来度量汉明距离$D_H(x_i,c'_i)$. 由此可得:

$$L_D=\frac{1}{K}\sum_{i}^{N}\sum_{k\in K}\left[c'_{i,k}\log h_{i,k}+\left(1-c'_{i,k}\right)\log\left(1-h_{i,k}\right)\right] \tag{8}$$

其中, K为预先设定的哈希编码位数, k为哈希编码的第k位.

在网络训练过程中, 生成的哈希码应向对应的哈希中心收敛, 然而输出编码为实数, 因此需要加入双峰拉普拉斯先验项, 对哈希编码进行二值约束, 强制哈希码向$\{-1,1\}$逼近, 减少二值化过程中的量化误差:

$$L_Q=\sum_{i}^{N}\left(\left\||2h_i-\mathbf{1}|-\mathbf{1}\right\|_1\right) \tag{9}$$

其中, $\mathbf{1}\in\mathbb{R}^K$是值全为 1 的向量, 其向量长度与哈希编码位数相等. 由于L_Q是一个不可导函数, 无法用于深度学习. 因此用$|x|\approx\log\cosh x$[30] 替代:

$$L_Q=\sum_{i}^{N}\sum_{k=1}^{K}\left(\log\cosh\left(\left|2h_{i,k}-1\right|-1\right)\right) \tag{10}$$

综上, 将两部分损失函数整理得到最终形式:

$$\min_{\Theta}L=L_D+\lambda L_Q \tag{11}$$

其中, Θ为深度哈希网络中所有参数的集合, λ为超参数, 实验中设置$\lambda=0.25$.

3 实验分析

3.1 实验环境

本次实验在一台机器上进行, 基于 CUDA 和 cuDNN 进行 GPU 加速, 机器配置如表 1 所列.

本次实验采用 Anaconda 4.11.0 进行包管理, 使用 PyTorch 1.7.1 实现所提模型.

3.2 实验数据集

将 SHNet 与当下主流的深度哈希方法在以下 4 个数据集上进行对比.

CIFAR-10: 是一个 10 类别的单标签图像数据集, 其中包括 50 000 张训练集图片, 10 000 张测试集图片. 实验中使用 1 000 张图片作为检索集, 5 000 张图片作为训练集, 剩余 54 000 张图片作为图像数据库.

表 1 实验机器配置信息

内容	配置信息
操作系统	Windows 10
处理器	AMD Ryzen 9 5900x@3.70 GHz
CPU核数	12
内存	64 GB
CUDA版本	11.4
cuDNN版本	8.2.2

ImageNet: 是一个 1 000 类别的单标签数据集, 该数据集包含 1.2M 以上的训练集图像和 50k 的训练集图像. 实验中从该数据集的训练集中取 100 个类别的所有图像数据作为本实验的图像数据库, 使用该数据库中的所有验证集图像作为检索集, 再从中每类各选取 100 张图片, 共 10 000 张图像作为训练集图像.

MS-COCO: 是一个 80 个类别的多标签图像数据集, 其中包括 82 783 张训练集图像, 40 504 张测试集图像, 实验中随机抽取 5 000 张图像作为检索集, 10 000 张图像作为训练集, 剩余的图像作为图像数据库.

NUS-WIDE: 是一个 81 类别的多标签分类数据集, 实验中随机选取 5 000 张图片作为检索集, 10 000 张图片作为训练集, 剩下的图片作为图像数据库.

3.3 评价指标

关于对比指标, 本文采用与 CSQ 论文中相同的 4 种评估指标: 平均准确率 mAP, 精度召回曲线 PR, 不同样本数量下检索精度曲线 P@N, 汉明距离在 2 以内的样本在不同 bits 下的检索精度曲线 P@H=2. 其中, 对于 CIFAR-10 数据集采用所有图片的检索结果计算 mAP, 对于 ImageNet 数据集采用 mAP@1000 (使用前 1 000 个检索结果计算 mAP), 对于 MS COCO 数据集和 NUS_WIDE 数据集则采用 mAP@5000 (使用前 5 000 个检索结果计算 mAP).

3.4 实验对比方法

为了验证 SHNet 在哈希图像检索任务上的优越性, 实验选取了该领域具有代表性的 7 种 state-of-the-art 深度哈希检索网络作为比较, 其中包括 CNNH, DNNH, DHN, DTSH, Hashnet, DCH, CSQ.

3.5 实验设置

在训练过程中, SHNet 采用用自适应矩估计 (Adam) 优化器, Adam 的初始学习率设置为 10^{-5}, 一阶矩估计的指数衰减率 β_1 设置为 0.9, 二阶矩估计的指数衰减率 β_2 设置为 0.009, 权重衰减参数设置为 0.005; 哈希编码层的学习率是特征层的 10 倍, 批归一化 (batchnorm) 的动量 (momentum) 设置为 0.1. 输入图像的批大小 (batchsize) 为 64.

为了实验公平的比较, 本文方法和所有对比方法在数据集上都采用相同的数据集划分策略和相同的预处理策略. 在网络架构上, 所有对比方法都采用了与本文一致的 ResNet50 网络结构作为骨架网络. 在训练策略上, 输入图片的批大小 (batchsize) 均为 64 张. 在训练参数上, 均采用原始参考文献中建议的最优参数作为训练.

3.6 结果分析

表 2 列出了 7 种深度哈希图像检索方法与 SHNet 在 4 个数据集上的不同哈希编码长度 (16 bits, 32 bits, 64 bits) 的平均准确率 (mAP) 结果.

表 2 8 种深度哈希方法在各数据集上的 mAP 结果

Method	CIFAR-10 (mAP@ALL)			ImageNet (mAP@1000)			MS COCO (mAP@5000)			NUS-WIDE (mAP@5000)		
	16 bits	32 bits	64 bits	16 bits	32 bits	64 bits	16 bits	32 bits	64 bits	16 bits	32 bits	64 bits
CNNH[10]	—	—	—	0.315	0.473	0.596	0.599	0.617	0.62	0.655	0.659	0.647
DNNH[11]	—	—	—	0.353	0.522	0.61	0.644	0.651	0.647	0.703	0.738	0.754
DHN[12]	0.838	0.859	0.865	0.367	0.522	0.627	0.719	0.731	0.745	0.712	0.759	0.771
DTSH[15]	0.805	0.833	0.846	0.652	0.718	0.829	0.671	0.71	0.733	0.763	0.792	0.816
Hashnet[16]	0.57	0.854	0.849	0.622	0.701	0.739	0.745	0.773	0.788	0.757	0.775	0.79
DCH[18]	0.668	0.694	0.678	0.652	0.737	0.758	0.759	0.801	0.825	0.773	0.795	0.818
CSQ[22]	0.826	0.844	0.84	0.842	0.874	0.878	**0.781**	**0.845**	0.875	0.81	0.828	0.844
SHNet (Ours)	**0.86**	**0.868**	**0.869**	**0.856**	**0.893**	**0.904**	0.765	0.842	**0.89**	**0.837**	**0.851**	**0.864**

从表 2 中可看出, 综合来说, 本文提出的 SHNet 要优于其他 7 种 state-of-the-art 深度哈希检索方法. 其中, 在 CIFAR 数据集上, 相较于其他 7 种方法, SHNet 取得了最高的 mAP, 用于对比的方法中表现最优的方法

是 DHN, SHNet 对比 DHN 在不同长度哈希码的 mAP 结果上分别取得了 2.2%、0.9%、0.4% 的 mAP 提升. 相较于 CSQ, SHNet 取得了 3.4%、2.4%、2.9% 的 mAP 提升. 在 ImageNet 数据集上, 相较于其他方法 SHNet 取得了最高的 mAP , SHNet 的 mAP 在不同长度哈希码上均优于 CSQ, 分别取得了 1.4%、1.9%、2.6% 的提升. 在 MS COCO 数据集上, CSQ 在的 16 bit 和 32 bit 上取得了最高的 mAP, 但在 64 bit 上, SHNet 相较于 CSQ 取得了 1.5% 的 mAP 提升. 在 NUS-WIDE 数据集上, SHNet 取得了最高的 mAP, 相较于 CSQ 方法, 分别取得了 2.7%、2.3%、2% 的 mAP 提升. 总体而言, 在 CIFAR-10, ImageNet, NUS-WIDE 这 3 个数据集上, SHNet 相较于 CSQ 在检索性能上平均提升了 2.9%、2%、2.2%.

为了进一步验证 SHNet 方法的有效性, 图 7 给出了 CNNH, DNNH, DHN, DTSH, Hashnet, DCH, CSQ 这 7 种深度哈希检索方法和 SHNet 方法在 ImageNet 数据集上的其他评价指标对比结果, 图 7(a) 给出了各种方法下的精度召回曲线图 (P-R), 图 7(b) 表示各种方法下前 1 000 个搜索结果的精度曲线图 (P@N), 图 7(c) 表示各种方法在汉明距离 D_H=2 时的检索精度折线图. 由图 7 中 3 个子图所示, 本论文提出的 SHNet 方法要显著优于其他 7 种深度哈希方法.

总的来说, 通过上述定量实验, 在相同的实验条件下, 实验得到的 SHNet 检索性能优于其他主流方法. 因为用以对比的主流深度哈希方法使用并行方式进行哈希编码, 忽略了编码自身的内在联系, 导致其编码性能受限. 而本文提出的 SHNet 方法所采用的串行哈希编码策略, 充分利用编码内部的关联性, 增强编码过程中的编码信息, 同时有效运用哈希码的冗余部分进行编码校验, 使得生成的哈希码更加紧凑, 检索性能更强.

图 8 中我们使用 t-SNE[31] 可视化 SHNet 和 CSQ 在 ImageNet 数据集上生成的 64 bits 哈希编码 (为了便于可视化, 我们只选取相同的 ImageNet 数据集的前 10 类数据进行可视化), 通过图 8 中数据分布可以看出, SHNet 相较于 CSQ 生成的哈希码具有更好的聚拢效果, 同类数据之间的分布被很好地控制在了哈希中心周围, 形成了明显的球形分布. 这表明了 SHNet 相较于 CSQ 有着更好的图像相似度检索性能. 这是由于 CSQ 所采用的并行的编码方式对应着高维度空间上的线性分类, 而 SHNet 采用多个子编码器级联实现串行编码, 该编码方式的机理是在低维度空间上进行多次线性分类的嵌套, 通过将高维度线性分界面分解为多个低维度线性空间的嵌套可实现更复杂的分界面, 从而充分利用维度冗余性与编码比特位关联性来提高模型性能的上限.

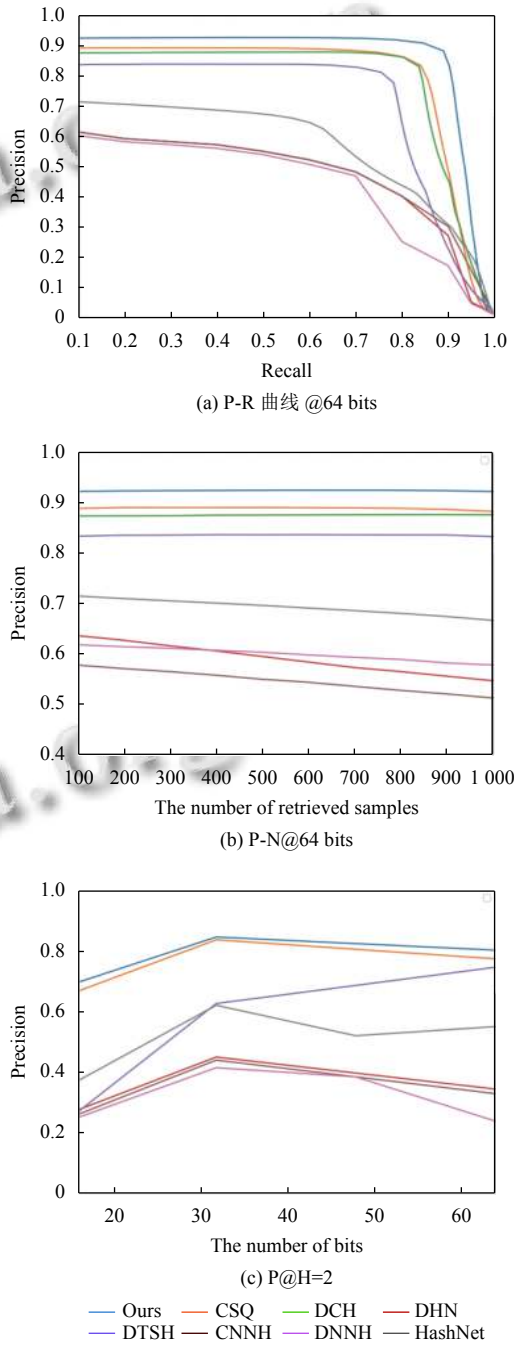


图 7 SHNet 与其他深度哈希方法在 ImageNet 上的评估对比

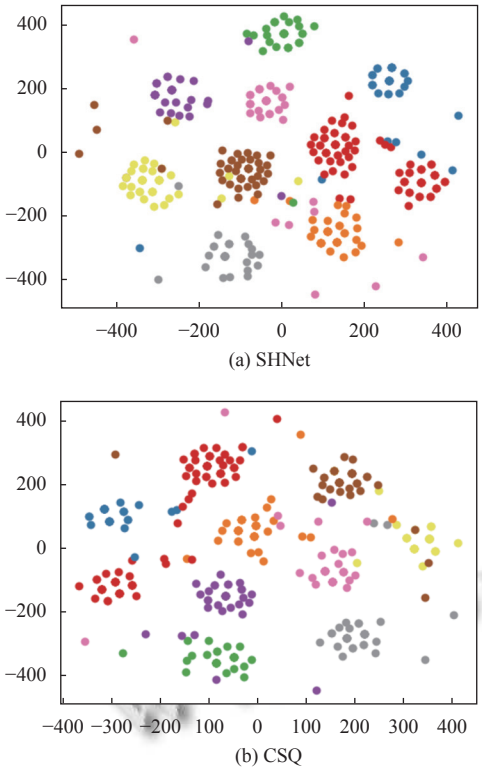


图 8 SHNet 和 CSQ 生成的哈希编码的 t-SNE 可视化

4 结论与展望

本文摒弃了传统深度哈希方法使用的并行编码方式, 提出了基于串行哈希编码的深度哈希方法 SHNet. SHNet 基于编码校验原理设计了一种新颖的串行校验编码的网络编码结构, 以串行编码方式生成哈希编码, 通过网络结构传递编码信息增强了编码内部的关联性并利用冗余编码位的校验能力提升编码性能. 本文以 4 个数据集 CIFAR-10、ImageNet、MS-COCO、NUS-WID 为实验数据集, 使用 CNNH, DNNH, DHN, DTSH, Hashnet, DCH, CSQ 这 7 种主流深度哈希检索方法和本文提出的 SHNet 方法进行 mAP 对比实验, 并且使用 t-SNE 可视化了 SHNet 和 CSQ 在 ImageNet 数据集上生成的哈希编码, 经以上实验结果证明, 串行哈希编码方式生成的哈希编码具有更好的检索性能, 在多个数据集上的检索效果优于其他 7 种主流深度哈希方法. 下一步的工作是将把串行校验编码方式应用到无监督的深度哈希图像检索领域.

参考文献

1 刘海龙, 李宝安, 吕学强, 等. 基于深度卷积神经网络的图像检索算法研究. 计算机应用研究, 2017, 34(12): 3816–3819. [doi: 10.3969/j.issn.1001-3695.2017.12.067]

2 Gionis A, Indyk P, Motwani R. Similarity search in high dimensions via hashing. Proceedings of the 25th International Conference on Very Large Data Bases. San Francisco: ACM, 1999. 518–529.

3 Shen FM, Shen CH, Liu W, *et al*. Supervised discrete hashing. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Boston: IEEE, 2015. 37–45.

4 Liu W, Wang J, Ji RR, *et al*. Supervised hashing with kernels. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Providence: IEEE, 2012. 2074–2081.

5 Gong YC, Lazebnik S, Gordo A, *et al*. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(12): 2916–2929. [doi: 10.1109/TPAMI.2012.193]

6 Jégou H, Douze M, Schmid C. Product quantization for nearest neighbor search. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(1): 117–128. [doi: 10.1109/TPAMI.2010.57]

7 Weiss Y, Torralba A, Fergus R. Spectral hashing. Proceedings of the 21st International Conference on Neural Information Processing Systems. Vancouver: ACM, 2008. 1753–1760.

8 Liu W, Wang J, Kumar S, *et al*. Hashing with graphs. Proceedings of the 28th International Conference on Machine Learning. Bellevue: ACM, 2011. 1–8.

9 Wang JD, Zhang T, Song JK, *et al*. A survey on learning to hash. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 40(4): 769–790. [doi: 10.1109/TPAMI.2017.2699960]

10 Xia RK, Pan Y, Lai HJ, *et al*. Supervised hashing for image retrieval via image representation learning. Proceedings of the 28th AAAI Conference on Artificial Intelligence. Québec City: ACM, 2014. 2156–2162.

11 Lai HJ, Pan Y, Liu Y, *et al*. Simultaneous feature learning and hash coding with deep neural networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Boston: IEEE, 2015. 3270–3278.

12 Zhu H, Long MS, Wang JM, *et al*. Deep hashing network for efficient similarity retrieval. Proceedings of the 30th AAAI Conference on Artificial Intelligence. Phoenix: AAAI, 2016. 2415–2421.

13 Liu HM, Wang RP, Shan SG, *et al*. Deep supervised hashing

for fast image retrieval. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 2064–2072.

14 Li WJ, Wang S, Kang WC. Feature learning based deep supervised hashing with pairwise labels. Proceedings of the 25th International Joint Conference on Artificial Intelligence. New York: ACM, 2016. 1711–1717

15 Wang XF, Shi Y, Kitani KM. Deep supervised hashing with triplet labels. Proceedings of the 13th Asian Conference on Computer Vision. Taipei: Springer, 2017. 70–84.

16 Cao ZJ, Long MS, Wang JM, *et al*. HashNet: Deep learning to hash by continuation. Proceedings of the IEEE International Conference on Computer Vision. Venice: IEEE, 2017. 5609–5618.

17 Li Q, Sun ZN, He R, *et al*. Deep supervised discrete hashing. Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach: ACM, 2017. 2479–2488.

18 Cao Y, Long MS, Liu B, *et al*. Deep cauchy hashing for hamming space retrieval. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 1229–1237.

19 Kang R, Cao Y, Long MS, *et al*. Maximum-margin hamming hashing. Proceedings of the IEEE/CVF International Conference on Computer Vision. Seoul: IEEE, 2019. 8251–8260.

20 Jiang QY, Li WJ. Asymmetric deep supervised hashing. Proceedings of the AAAI Conference on Artificial Intelligence. New Orleans: AAAI, 2018. 409.

21 Zhang Z, Zou Q, Lin YW, *et al*. Improved deep hashing with soft pairwise similarity for multi-label image retrieval. IEEE Transactions on Multimedia, 2020, 22(2): 540–553. [doi: 10.1109/TMM.2019.2929957]

22 Yuan L, Wang T, Zhang XP, *et al*. Central similarity quantization for efficient image and video retrieval. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 3080–3089.

23 Hoe JT, Ng KW, Zhang TY, *et al*. One loss for all: Deep hashing with a single cosine similarity based learning objective. Proceedings of the 35th Conference on Neural Information Processing Systems. NeurIPS, 2021. 24286–24298.

24 Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images. Technical Report, University of Toronto, Toronto. 2009.

25 Russakovsky O, Deng J, Su H, *et al*. Imagenet large scale visual recognition challenge. International Journal of Computer Vision, 2015, 115(3): 211–252. [doi: 10.1007/s11263-015-0816-y]

26 Lin TY, Maire M, Belongie S, *et al*. Microsoft COCO: Common objects in context. Proceedings of the 13th European Conference on Computer Vision. Zurich: Springer, 2014. 740–755.

27 Chua T S, Tang J, Hong R, *et al*. NUS-WIDE: A real-world Web image database from National University of Singapore. Proceedings of the ACM International Conference on Image and Video Retrieval. Santorini: ACM, 2009. 48.

28 He KM, Zhang XY, Ren SQ, *et al*. Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 770–778.

29 Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. Proceedings of the 32nd International Conference on Machine Learning. Lille: ACM, 2015. 448–456.

30 Hyvärinen A, Hurri J, Hoyer PO. Natural Image Statistics: A Probabilistic Approach to Early Computational Vision. New York: Springer, 2009.

31 Donahue J, Jia YQ, Vinyals O, *et al*. DeCAF: A deep convolutional activation feature for generic visual recognition. Proceedings of the 31st International Conference on Machine Learning. Beijing: JMLR, 2014. 647–655.

(校对责编: 牛欣悦)