arXiv:2305.04710v1 [cs.CV] 8 May 2023

# *ElasticHash*：基于 Elasticsearch 的深度哈希语义图像相似性搜索

Nikolaus Korfhage、Markus M¨uhling 和 Bernd Freisleben

数学与计算机科学系，马尔堡大学，马尔堡，德国{
korfhage,muehling,freisleb}@informatik.uni-marburg.de

**摘要**。 我们提出了 *ElasticHash*，这是一种新颖的方法，旨在实现高质量、高效率和大规模的语义图像相似性搜索。该方法基于深度哈希模型，用于学习哈希码，以便在自然图像中进行细粒度的图像相似性搜索；同时采用两阶段方法，利用 Elasticsearch (ES) 高效搜索二进制哈希码。在第一阶段，使用多索引哈希和 ES 邻近哈希码词项查找，基于较短的哈希码执行粗略搜索。在第二阶段，通过计算较长哈希码的汉明距离，对结果列表进行重新排序。我们在 OpenImages 数据集上进行了评估，使用超过 120,000 张查询图像，在约 690 万张数据库图像上测试了 *ElasticHash* 的检索性能。结果表明，我们的方法能够实现高质量的检索效果，同时保持较低的搜索延迟。

**关键词：** 深度哈希 · 相似性搜索 · Elasticsearch

## 1 引言

基于深度卷积神经网络（CNNs）学习得到的特征表示的基于内容查询方法，显著提升了基于内容的图像检索系统的性能。然而，语义图像相似性搜索领域的最先进方法，受限于浅层网络架构，以及训练和评估阶段图像类别较少的小型数据集。训练阶段图像类别较少，会导致模型难以泛化到评估阶段内容未知的查询图像；换言之，需要对图像内容进行更细粒度的建模。因此，要满足用户在科学研究和商业应用中的诸多需求，必须具备对任意搜索查询的高准确率、快速响应时间，以及扩展至数百万张图像的能力。

本文提出*ElasticHash*，这是一种高质量、高效且可扩展的方法，基于最流行的企业级全文搜索与分析引擎 Elasticsearch[1] (ES)，实现语义图像相似性搜索。ES 借助基于 Lucene[2]，的倒排索引，能够极快地处理查询，并可扩展至数百

[1] https://www.elastic.co[2]
https://lucene.apache.org

台服务器，提供负载均衡，并支持高可用性和可靠性。显然，ES 的这些特性不仅适用于全文搜索，也适用于语义图像相似性搜索。此外，将图像相似性搜索集成到 ES 中，还能支持多模态查询，例如在单个查询中结合文本和图像。本文的贡献如下：

– 我们提出了 $ElasticHash$，这是一种新颖的两阶段语义图像相似性搜索方法，基于多索引哈希，并通过词项查找查询将其集成到 ES 中。– 我们展示了实验结果，表明$ElasticHash$ 能够同时实现快速响应时间和高质量检索结果，这是通过结合短哈希码（搜索速度更快）和长哈希码（检索质量更高）的优势来实现的。据我们所知，我们首次在 OpenImages 数据集的约 690 万张数据库图像上，对超过 12 万张查询图像进行了图像相似性搜索评估。– 我们在 http://github.com/umr-ds/ ElasticHash 提供深度图像相似性搜索模型、相应的 ES 索引以及演示应用。

本文组织如下：第 2 节讨论相关工作。第 3 节介绍$ElasticHash$。第 4 节从搜索延迟和检索质量两个方面评估$ElasticHash$ 在 OpenImages 数据集上的表现。第 5 节总结本文并展望未来工作方向。

## 2 相关工作

深度学习，尤其是深度卷积神经网络，显著提升了基于内容的图像相似性搜索性能。随着底层图像数据库规模不断扩大，对高效相似性搜索策略的需求也随之增加。由于高维卷积神经网络特征不适合在超大规模数据库中进行高效搜索，大规模图像相似性搜索系统转而采用二进制图像编码进行量化或紧凑表示，并通过快速比较实现搜索，而非直接使用完整的卷积神经网络特征。

近期，已有多种深度哈希方法被提出 [25,6,23,21,2,15,4]。其中许多方法采用成对损失或三元组损失。尽管这些方法通常在其测试数据集上取得最先进的性能，但未必适用于超大规模数据集，以及基于数千个类别的细粒度图像相似性搜索。现有的深度哈希方法通常使用较小的卷积神经网络进行训练，而这类网络往往无法捕捉超大规模图像数据集中的细粒度特征。它们通常采用 AlexNet [12] 等 CNN 模型作为骨干网络，并且通常只在少量图像类别上进行评估 [22,4,6,25]（例如，ImageNet 中抽取的 100 个类别 [4], 、COCO 中约 80 个物体类别[14], 、包含 81 个概念的 NUS-WIDE [5] ，甚至是 MNIST 或 CIFAR 中仅有的 10 个类别）。此外，CIFAR 和 MNIST 中的图像尺寸非常小（分别为 32x32 和 28x28），因此不足以满足真实应用中的图像相似性搜索需求。许多方法是在规模相对较小的

训练数据集上训练的（例如，10,000 - 50,000 张图像 [3,4,15]）。此外，目前没有标准化的基准数据集，各篇论文使用的训练、查询和数据库图像划分各不相同，这进一步增加了方法比较的难度。而且，对于大型数据集，从头开始训练的成本可能高得令人难以承受。我们发现，对于图像类别数量众多的大型数据集，结合三元组损失与分类损失的迁移学习方法能够取得良好的检索效果。据我们所知，*ElasticHash* 是首个提出在数量足够多的图像类别上训练并评估深度哈希模型的工作。

目前，图像表征学习哈希的最佳方法主要分为乘积量化（PQ）方法 [8,9] 和基于深度哈希（DH）的方法 [23,6]。Amato 等人 [1] 提出了 PQ 方法，可将神经网络特征转换为适合在 ES 中建立索引的文本格式。然而，这种方法无法达到 FAISS [9]的检索性能。因此，我们专注于深度哈希，并将其与多索引哈希（MIH） [17] 结合，从而避免在汉明空间中进行穷举搜索，同时在保持高检索质量的情况下实现较低的搜索延迟。

*ElasticHash* 与其他集成到 ES 中的图像相似性搜索方法有关。例如，FENSHSES [16] 将 MIH 集成到 ES 中，其搜索延迟与 FAISS 相当。该方法在汉明球半径较小、数据集规模相对较小时（50 万张图像）运行效率很高。然而，较小的汉明半径往往会导致查询 [17]得到的近邻过少。因此，像 FENSHSES 这样的 MIH 并不适用于我们的场景：在 ES 中使用较长的二进制编码（256 位）进行大规模图像检索，并要求在约 700 万张图像的数据集上实现亚秒级搜索延迟。此外，我们仅使用部分比特而非完整哈希码进行基于 MIH 的粗略搜索，从而解决了 FENSHSES 的不足。其他研究通过修改 Lucene 库 [7], 来扩展 ES 的图像相似性搜索功能，而我们的方法则无需修改 ES 代码库即可无缝集成。

## 3 *ElasticHash*

*ElasticHash* 由多个组件构成，如图 1 所示：深度哈希组件、ES 集群和检索组件。深度哈希组件通过 TensorFlow Serving 实现为一项 Web 服务，集成的深度哈希模型会对图像进行处理，并返回相应的二进制编码。在第一阶段，使用深度哈希组件从索引阶段的数据库图像中提取二进制编码，并将其存储到 ES 集群中。初始索引构建完成后，检索组件会处理传入的查询图像，并将检索结果可视化。为此，系统通过 Web 服务从查询图像中提取二进制编码，组装相应的 ES 查询并发送至 ES 集群，由其返回最终的相似图像列表。整个相似性搜索系统可通过 Docker 轻松部署到生产环境。

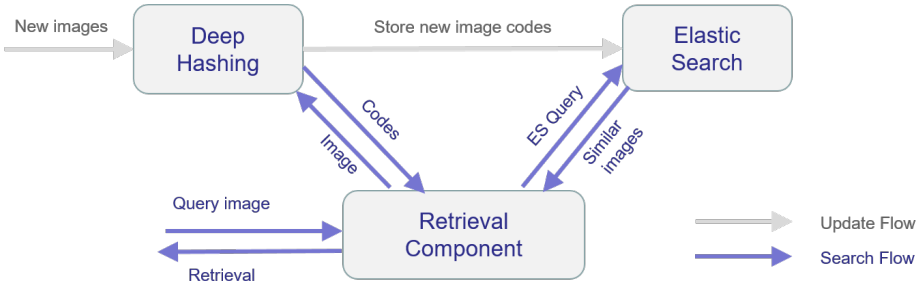


**图 1.** ES 中图像相似性搜索工作流程概览。

第 3.1 节将更详细地介绍深度哈希模型，包括训练策略和网络架构。第 3.2 节将介绍 ES 集成。

### 3.1 深度哈希模型

现在，我们将介绍我们的深度哈希模型，以及如何使用该模型提取短哈希码和长哈希码。

模型训练分为两个阶段，均使用 ADAM 作为优化方法。首先，使用包含更多类别的数据集训练一个在 ImageNet 上预训练的 EfficientNetB3 [19] 模型，以获得更细粒度的嵌入。与原始 ImageNet 数据集相比，该数据集包含所有训练图像超过 1000 张的 ImageNet 类别，以及 Places2 [24] 数据集中的所有类别，类别总数达到 5,390。模型使用 Softmax 输出和交叉熵损失进行训练。先以 0.01 的学习率训练最后一层两个 epoch，然后以 0.0001 的学习率训练所有层 16 个 epoch。

在第二阶段，使用分类模型的权重初始化深度哈希模型。该模型在类别输出层之前增加了一个 256 位编码层，该层使用 $tanh$ 激活函数，输出维度为 256。模型以 0.0001 的学习率训练 5 个周期。训练数据集与之前相同，但结合了输出层上的交叉熵损失和编码层上的困难三元组损失 [18]。

结合分类损失

$$\mathcal{L}_c = \sum_{i=1}^{K} y_i \log p_i \tag{1}$$

对于 $K$ 个类别及其标签 $y_i$ 和预测结果 $p_i$，以及三元组损失

$$\mathcal{L}_t = max(d(a,p) - d(a,n) + \gamma, 0) \tag{2}$$

对于锚图像 $a$ 与正样本 $p$ 的编码层 256 维输出之间的欧几里得距离 $d$，以及 $a$ 与负样本之间的距离

$n$，组合损失函数分别定义为：

$$\mathcal{L} = \alpha\mathcal{L}_{\rfloor} + \beta\mathcal{L}_{\sqcup}, \tag{3}$$

其中我们设置边界 $\gamma = 2$ 以及权重 $\alpha = 1$ 和 $\beta = 5$。我们首先从各类别的均匀分布中采样一个包含 $b = 128$ 张图像的批次。该批次既用于计算分类损失，也用于生成 $b$ 个困难三元组。为了使相似性搜索更加稳健，我们在这两个阶段都采用了大量数据增强；除了标准的数据增强方法外，还包括引入 JPEG 压缩伪影。训练完成后，该模型会生成 256 位编码。这些编码可以分解为四个 64 位编码，以便在长整数上快速计算汉明距离。然而，在一个包含约 1,000 万张图像的数据集上使用如此长度的编码成本过高，即使采用多索引哈希也是如此。因此，我们从原始的 256 位编码中提取 64 位编码，以便在更短的编码上执行过滤，从而使用更小的汉明球半径。为了从 256 位编码中提取最重要的 64 个比特，我们首先根据比特之间的相关性，应用 Kernighan-Lin 算法 [10]将 256 位编码划分为四个分区。然后，从这四个去相关的分区中分别取前 16 个比特，组成 64 位编码。

### 3.2 集成到 ES

在介绍我们将图像相似性搜索集成到 ES 之前，我们先简要回顾一下汉明空间中的多索引哈希 [17]。多索引哈希的思路基于以下观察：对于两个二进制编码 $h = (h^1, \ldots, h^m)$ 和 $g = (g^1, \ldots, g^m)$，其中$m$ 为分区数，$h^k$ 和 $g^k$ 是 $k^{th}$ 子码，$H$ 是汉明范数，以下命题成立：

$$\|h - g\|_H \leq r \Rightarrow \exists k \in \{1, \ldots, m\} \ \left\|h^k - g^k\right\|_H \leq \left\lfloor \frac{r}{m} \right\rfloor \tag{4}$$

对于将 64 位编码分解为 $m = 4$ 个子码的情况，这意味着：如果至少有一个子码与查询子码的距离为 $d \leq \lfloor \frac{r}{m} \rfloor = 2$，那么该编码就位于汉明半径 $r < 12$ 内。如果各子码之间的独立性最大化，多索引哈希的性能就能得到提升 [20]，尤其是对于较短的编码 [16]。因此，在训练深度哈希模型后，应相应地重新排列各比特的位置。

用于检索的 ES 索引中，每张图像包含四个短码（f 0 - f 3）和四个长子码（r 0 - r 3）。短码用于多索引哈希（MIH），能够高效利用 ES 的倒排索引结构，因此被拆分为四个类型为“keyword”的子码。长码也被拆分为四个子码，以便快速计算 long 类型值的汉明距离。

另外还使用了一个索引，以便在检索查询中快速查找相邻子码。近邻索引创建后便不会再改变，仅用作词项查询的辅助索引。它需要预先计算所有可能的 16 位子码各自的全部最近邻。因此，近邻索引

包含 $2^{16}$ 个文档。文档 ID 对应于 16 位子码的无符号整数表示，因此可以通过词项查询访问。它仅包含一个名为 ”nbs” 的字段，该字段被赋值为一个列表，其中列出了与对应查询子码的汉明距离在 $d$ 以内的所有相邻 16 位码。由于该索引本质上用作查找表，因此也可以在其他位置实现，即不必将其作为 ES 索引。不过，以这种方式集成查找表，无需外部代码即可实现，并能快速部署整个系统。所有表示可能的 16 位子码的文档，均按照清单 1.1 中的查询插入。

```
POST /nbs-d2/_doc/<16 bit subcode>
{ "nbs" : [ <d2 neighbors of 16 bit subcode>  ] }
```

**清单 1.1**。 向邻居查找索引添加条目的查询。

在此阶段，通过查询邻居的附加索引来实现多索引哈希，从而快速进行邻居查找。即使使用多索引哈希，对于较大的数据库，采用针对 256 位编码训练的深度哈希模型的完整编码长度，计算成本仍然过高。因此，我们将过滤阶段的编码长度限制为 64 位编码。为了在第一阶段获得足够大的候选哈希码集合，我们需要在半径相应较大的汉明球内进行搜索。我们设定 $d = 2$，这将至少返回距离某个 64 位编码不超过 $r = 11$ 的所有编码。在我们的设置中，$d = 2$ 会为每个子码产生 137 个近邻，即总计 548 个近邻。

在 ES 中，我们通过词项查找来实现多索引哈希。它会获取现有文档的字段值，然后将这些值用作搜索词项（见清单 1.1）。与将所有近邻都放入查询中相比，为子码邻居使用专用索引的优势在于，近邻子码的检索是在 ES 内部完成的。因此，查询负载较小，也不需要在外部处理邻居查找。

在第二阶段，根据各代码与长代码之间的汉明距离，对通过多索引哈希获得的所有代码进行重新排序。为计算 256 位代码的汉明距离，将清单 1.2 中的 Painless 脚本应用于四个子码中的每一个。

```
POST _scripts/hd64
{ "script": { "lang": "painless",
  "source": 64-Long.bitCount(params.subcode^doc[params.field].value) } }
```

**清单 1.2**。 用于添加 Painless 脚本的查询。

清单 1.3 中的查询将多索引哈希步骤作为过滤器与词项查询相结合，并将重新排序步骤作为对过滤后的检索列表应用清单 1.2 中的 Painless 脚本。

## 4 实验评估

为了确定*ElasticHash*的搜索延迟和检索质量，我们评估了三种设置，考察如何在 ES 中使用深度哈希模型生成的二进制哈希码进行大规模图像检索：(1) 短码，即过滤和重新排序均使用 64 位码；

```
GET /es-retrieval/_search
{ "query": {
"function_score": {
 "boost_mode": "sum",  "score_mode": "sum",
 "functions": [ ..., {
  "script_score": {
   "script": { "id": "hd64",
    "params": {
     "field": "r_<i>",
     "subcode":  <64 bit subcode for re-ranking> } } },  "weight": 1 },  ... ],
"query": {
  "constant_score": {
   "boost": 0.
   "filter": {
    "bool": {
     "minimum_should_match": 1,
     "should": [ ..., {
      "terms": {
       "f_<j>": {
        "id": "<16 bit subcode for lookup>",
        "index": "nbs-d2",
        "path": "nbs" } } },  ... ] } } } },  } } }
```

**清单 1.3**。 用于执行两阶段相似度搜索的查询。

(2) 长码，即过滤和重新排序均使用 256 位码；以及 (3) *ElasticHash*，即过滤使用 64 位码，重新排序使用 256 位码。设置 (1) 和 (2) 与 Mu 等人 [16] 的多索引哈希集成方案类似。

为了评估我们的方法，我们使用 OpenImages [13], ，它是目前公开可用的规模最大的带标注图像数据集。该数据集包含 920 万张 Flickr 图像的多标签标注，涵盖 19,794 个不同标签，并划分为训练集、验证集和测试集。训练集平均每张图像有 2.4 个正标签，而验证集和测试集平均有 8.8 个。我们将下载数据集时可获取的所有训练图像用作数据库图像，总计 6,942,071 张。为了评估检索质量，我们将从 OpenImages 测试集和验证集中下载的所有图像用作查询图像（共 121,588 张）。我们从这些图像中抽取 10,000 张，以测量三种不同设置下的搜索延迟。

检索列表的质量使用平均精度（AP）评分进行评估，这是图像检索中最常用的质量度量指标。AP 评分根据检索到的图像列表按如下方式计算：

$$AP(\rho) = \frac{1}{|R \cap \rho^N|} \sum_{k=1}^{N} \frac{\left|R \cap \rho^k\right|}{k} \psi(i_k), \text{with} \quad \psi(i_k) = \begin{cases} 1 & \text{if } i_k \in R \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

其中， $N$是排序后图像列表的长度， $\rho^k = \{i_1, i_2, \ldots, i_k\}$是截至排名 $k$的排序后图像列表， $R$是相关文档的集合，$\left|R \cap \rho^k\right|$是 $\rho$ 中排名前 $k$ 的图像里相关图像的数量， $\psi(i_k)$是相关性函数。如果某张图像至少有一个标签与查询图像相同，我们就将其视为相关图像。为评估整体性能，我们计算所有查询的 AP 分数的平均值，得到平均平均精度分数。




**图2**。 同一查询图像（左侧第一张）的（a）短码、（b）长码和（c） *ElasticHash* 前10名检索结果；绿色：相关结果；红色：不相关结果。

### 4.1 结果

我们首先评估查询的搜索延迟。接着，我们根据平均精度（AP）比较检索质量。实验在一台配备 Intel Core i7-4771 中央处理器（3.50 GHz）和 32 GB 随机存取存储器的系统上进行。

**表1**。 在121,588张查询图像上，不同 $k$ 阈值下的平均平均精度检索质量。

| 前 $k$ | 10 | 25 | 50 | 100 | 250 | 500 | 1000 |
|---|---|---|---|---|---|---|---|
| 短 | 87.94 | 86.08 | 84.44 | 82.54 | 79.41 | 76.44 | 72.86 |
| long | 95.35 | 94.72 | 94.23 | 93.71 | 92.90 | 92.09 | 90.95 |
| *ElasticHash* | 95.21 | 94.48 | 93.90 | 93.22 | 92.02 | 90.61 | 88.42 |

**表 2.** 在 10,000 张查询图像上，针对 $k$ 的不同阈值，ES 查询的搜索延迟（毫秒）及标准差。

| 前 $k$ | | 10 | 25 | 50 | 100 | 250 | 500 | 1000 |
|---|---|---|---|---|---|---|---|---|
| 简短 | $\mu$ | 23.09 | 23.98 | 24.45 | 25.58 | 28.38 | 33.09 | 42.20 |
| | $\sigma$ | 4.74 | 4.65 | 4.70 | 4.72 | 4.86 | 5.20 | 6.07 |
| long | $\mu$ | 111.83 | 111.58 | 111.99 | 113.05 | 116.77 | 121.98 | 132.60 |
| | $\sigma$ | 16.50 | 16.58 | 16.72 | 16.54 | 17.04 | 17.13 | 17.99 |
| *ElasticHash* | $\mu$ | 36.12 | 36.75 | 37.28 | 38.17 | 40.88 | 45.73 | 55.23 |
| | $\sigma$ | 7.80 | 7.96 | 7.81 | 7.89 | 7.93 | 8.12 | 8.64 |

表1显示，当 $k$ 不超过250时，相较于两个阶段都使用长码，采用 *ElasticHash* 并不会显著降低检索质量。图2展示了三种设置下前10名检索结果的示例。显然，*ElasticHash* 的检索质量与使用

长码相近，而且两者都优于使用短码。另一方面，表2表明，与两个阶段都使用短码相比，平均检索时间仅略有增加。这说明*ElasticHash* 在检索质量和搜索延迟之间实现了良好的权衡。尽管我们的深度哈希模型是在5,390个类别上训练的，但验证数据集中几乎包含20,000个类别，*ElasticHash*仍取得了较高的平均精度（AP）。

## 5 结论

我们提出了 *ElasticHash*，这是一种新颖的两阶段语义图像相似性搜索方法，基于深度多索引哈希，并通过 terms lookup 查询集成到 ES 中。我们在大型图像数据集上的实验结果表明，利用短哈希码（搜索速度更快）和长哈希码（检索质量更高）的优势，可以同时实现较低的搜索延迟和高质量的检索结果。

未来工作有几个方向。例如，研究需要多少个类别才能获得较强的泛化能力，将会很有意义。此外，我们可以将损失函数调整为适用于多标签图像数据。最后，我们计划扩展该方法，以便使用 ES 实现意图图像相似性搜索 [11]。

## 6 致谢

本研究得到德国研究基金会（DFG 项目编号 388420599）和 HMWK（LOEWE 研究集群 Nature 4.0）的资助。

## 参考文献

1. Amato, G., Bolettieri, P., Carrara, F., Falchi, F., Gennaro, C.：使用 Elasticsearch 进行大规模图像检索。载于：第 41 届 ACM SIGIR 信息检索研究与发展国际会议。第 925–928 页（2018） 2. Cao, Y., Qi, H., Zhou, W., Kato, J., Li, K., Liu, X., Gui, J.：面向大数据近似最近邻搜索的二进制哈希：综述。IEEE Access **6**，2039–2054（2017） 3. Cao, Z., Long, M., Wang, J., Yu, P.S.：HashNet：通过连续化进行深度学习哈希。载于：IEEE 计算机视觉国际会议论文集。第 5608–5617 页（2017） 4. Cao, Z., Sun, Z., Long, M., Wang, J., Yu, P.S.：深度优先级哈希。载于：第 26 届 ACM 多媒体国际会议论文集。第 1653–1661 页（2018） 5. Chua, T.S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.：NUS-WIDE：新加坡国立大学的真实网络图像数据库。载于：ACM 图像与视频检索国际会议论文集。第 1–9 页（2009） 6. Erin Liong, V., Lu, J., Wang, G., Moulin, P., Zhou, J.：用于学习紧凑二进制编码的深度哈希。载于：IEEE 计算机视觉与模式识别会议论文集。第 2475–2483 页（2015）

7. Gennaro, C., Amato, G., Bolettieri, P., Savino, P.：一种基于 Lucene 搜索引擎库的内容图像检索方法。载于：第 14 届欧洲数字图书馆研究与先进技术会议论文集。第 55–66 页。施普林格（ 2010 ） 8. Jegou, H., Douze, M., Schmid, C.：用于最近邻搜索的乘积量化。IEEE 模式分析与机器智能汇刊 **33**(1)，117–128（2010） 9. Johnson, J., Douze, M., J´egou, H.：借助 GPU 进行十亿级相似性搜索。IEEE 大数据汇刊（2019） 10. Kernighan, B.W., Lin, S.：一种高效的图划分启发式过程。贝尔系统技术杂志 **49**(2)，291–307（1970） 11. Korfhage, N., M¨uhling, M., Freisleben, B.：意图图像相似性搜索。载于：IAPR 模式识别人工神经网络研讨会。第 23–35 页。施普林格（ 2020） 12. Krizhevsky, A., Sutskever, I., Hinton, G.E.：使用深度卷积神经网络进行 ImageNet 分类。神经信息处理系统进展**25**，1097–1105（2012） 13. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Ka- mali, S., Popov, S., Malloci, M., Kolesnikov, A., 等：Open Images 数据集 V4。国际计算机视觉杂志，第 1–26 页（2020） 14. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Doll´ar, P., Zitnick, C.L.：Microsoft COCO：上下文中的常见物体。载于：欧洲计算机视觉会议。第 740–755 页。施普林格（2014） 15. Liu, H., Wang, R., Shan, S., Chen, X.：用于快速图像检索的深度监督哈希。载于：IEEE 计算机视觉与模式识别会议论文集。第 2064–2072 页（2016） 16. Mu, C.M., Zhao, J.R., Yang, G., Yang, B., Yan, Z.J.：在全文搜索引擎中进行汉明空间快速、精确的最近邻搜索。载于：相似性搜索与应用国际会议。第 49–56 页。施普林格（2019） 17. Norouzi, M., Punjani, A., Fleet, D.J.：利用多索引哈希在汉明空间中快速搜索。载于：2012 年 IEEE 计算机视觉与模式识别会议（CVPR）。第 3108–3115 页。IEEE（2012） 18. Schroff, F., Kalenichenko, D., Philbin, J.：FaceNet：用于人脸识别与聚类的统一嵌入。载于：IEEE 计算机视觉与模式识别会议论文集。第 815–823 页（2015） 19. Tan, M., Le, Q.：EfficientNet：重新思考卷积神经网络的模型缩放。载于：国际机器学习会议。第 6105–6114 页（2019） 20. Wan, J., Tang, S., Zhang, Y., Huang, L., Li, J.：数据驱动的多索引哈希。载于：2013 年 IEEE 国际图像处理会议。第 2670–2673 页（2013） 21. Wang, J., Zhang, T., Song, J., Sebe, N., Shen, H.T.：学习哈希综述。IEEE 模式分析与机器智能汇刊 **40**(4)，769–790（2018） 22. Wang, J., Kumar, S., Chang, S.F.：面向大规模搜索的半监督哈希。IEEE 模式分析与机器智能汇刊 **34**(12)，2393–2406（2012） 23. Wang, J., Liu, W., Kumar, S., Chang, S.F.：用于大数据索引的学习哈希：综述。IEEE 会议录 **104**(1)，34–57（2015） 24. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.：Places：用于场景识别的千万级图像数据库。IEEE 模式分析与机器智能汇刊 **40**(6)，1452–1464（2017） 25. Zhu, H., Long, M., Wang, J., Cao, Y.：用于高效相似性检索的深度哈希网络。载于：AAAI 人工智能会议论文集。第 30 卷（2016）