文章编号:1671-4598(2024)06-0286-06 DOI:10.16526/j.cnki.11-4762/tp.2024.06.041 中图分类号:TP391 文献标识码:A

基于层次聚类的图文检索模型研究

孙健玮，刘玉龙
（中国电子科技集团公司 第15研究所，北京 100083）

摘要：图文检索在工业中的用途和作用是多方面的，可以帮助提高研发和生产效率，促进科技创新，提高产品的质量和竞争力；目前，图文检索模型的重点是提高检索的精度；随着技术和数据的快速发展，深度学习和大模型技术的不断应用，图文检索的速度问题逐渐凸显，为解决当前图文检索速度受限、计算量大的问题，提出了一种基于层次聚类的图文检索模型；该方法选择了检索效果明显的跨模态哈希方法，并运用深度聚类算法对待检索的数据进行分类，从而缩小检索范围，提高了检索速度；实验结果表明，基于层次聚类的图文检索模型在保持检索精度的同时，显著提高了检索速度，使得工程人员能够更快地获取到满意的检索结果。

关键词：图文检索；跨模态哈希方法；深度学习；深度聚类算法；信息检索

Research on Image and Text Retrieval Model Utilizing Hierarchical Clustering

SUN Jianwei, LIU Yulong
(15th Research Institute, China Electronics Technology Group, Beijing 100083, China)

Abstract: The application and impact of image-text retrieval in industry are multifaceted, as it can help improve research and development efficiency, promote technological innovation, and enhance product quality and competitiveness. Currently, the emphasis of image-text retrieval models is on improving retrieval accuracy. With the rapid development of technology and data, the continuous application of deep learning and large-scale model techniques has gradually highlighted the issue of retrieval speed in image-text retrieval. To address the current limitations in retrieval speed and high computational requirements, a hierarchical clustering-based image-text retrieval model has been proposed. This method adopts a cross-modal hashing approach with evident retrieval effectiveness and applies deep clustering algorithms to classify the data to be retrieved, thereby reducing the retrieval scope and improving retrieval speed. Experimental results indicate that the hierarchical clustering-based image-text retrieval model significantly enhances retrieval speed while maintaining retrieval accuracy, enabling engineering personnel to obtain satisfactory retrieval results more quickly.

Keywords: image-text retrieval; cross-modal hashing; deep learning; deep clustering; information retrieval

0 引言

图文检索[1-2]是信息检索领域的一个重要分支，图文检索在工作及工业生产中能够快速准确地查找和管理各种信息资源，提高工作效率，优化决策流程，同时帮助提升产品设计与研发、质量控制、设备维护等方面的效能，为工作和生产带来便利与智能化体验。随着信息技术和人工智能的发展，图文检索也经历了从传统的基于关键词的检索到基于内容的图像检索，发展为跨模态检索[3-7]。近年来随着深度学习和多模态学习的兴起，跨模态检索成为研究热点，跨模态检索能够处理文本、图像、视频等多种形式的信息，通过深度学习模型实现语义理解和跨模态匹配，提高了检索的准确性和多样性。

为了面向实际需求，跨模态检索的实时性要求需要跨模态检索在效率上有更高的结果。将哈希法[8-11]与深度学习相结合，多媒体信息检索的准确性和效率得到显著提高。获取图像的空间依赖性和文本的时间动态是学习潜在特征表示和跨模态关系的重要任务，因为它减少了模态之间的异质性差距。因此，文献[12]提出了一种新的深度视觉语义哈希模型。它在一个完整的深度学习架构中创建文本句子和图像的简洁哈希码，捕捉自然语言和视觉数据之间的基本跨模态对应。文献[13]将哈希码学习和特征学习结合到同一个框架中，引入了一种新的深度跨模态哈希技术。从头到尾，这个框架由深度神经网络组成，每个模式一个，从一开始就进行特征学习。文献[14]提出了一个自监督的语义网络，这个网络针对多标签信息进一步挖掘高层语义信息，使用得到的语义信息作为监督信息来指导不同模态的特征学习过程，以此，模态间的相似关系可以同时在共同语义空间和汉明空间内得以保持，有效地减小了模态之间的差异，进而产生精确的哈希码，提高检索精度。文献[15]提出了一种基于三元组的深度哈希网络。在文献[16]中，提出了一种新的跨模态零 shot 哈希方法，该方法有效地利用了具有独立标签空间的标记和非标记多模态数据。

随着跨模态检索数据量和种类的不断扩张，检索效率

收稿日期:2024-05-21； **修回日期**:2024-05-27。

作者简介:孙健玮(1998-),男,硕士。

引用格式:孙健玮,刘玉龙.基于层次聚类的图文检索模型研究[J].计算机测量与控制,2024,32(6):286-291,298.

成为当前跨模态检索面临的重要挑战之一。大规模、多样化的数据使得传统的检索方法难以有效处理，导致检索范围广泛且匹配计算量巨大。

本文通过研究分析跨模态哈希检索面临的困难和挑战，提出了基于层次聚类的图文检索模型，该模型通过多模态 Transformer[17] 模型提取图像和文本的特征，并通过压缩形成哈希码，然后通过引入无监督聚类方法对跨模态哈希码进行聚类，将相似的哈希码分配到同一簇中。通过聚类，将复杂冗余的检索范围进行分层分类，并根据聚类簇的结果来决定，检索数据计算的范围。具有距离相近的哈希码将在一定范围内进行哈希检索，从而减少跨模态检索的数据总量，提高跨模态哈希检索的效率。这种方法通过将相似的哈希码进行聚类和降维，有效地减少了计算和存储的开销，同时保持了较高的检索准确性和效率。

1　模型架构及原理

1.1　本文图文检索框架

图文检索模型架构如图 1 所示。主要由数据层、特征提取层、特征对齐层、特征压缩层、聚类层和检索层 6 个部分组成。

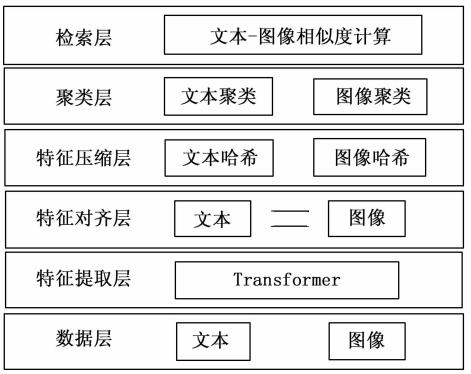

图 1　图文检索模型框架图

首先需要对文本和图像数据进行特征提取，这里选择近年来，特征提取与学习优秀的模型 Transformer[17]。Transformer 模型具有可扩展性，可以通过增加层和注意力头的方式来增加模型的深度和复杂度，以适应更复杂的多模态数据建模任务。此外，Transformer 模型的结构也非常灵活，可以通过添加或修改模块来适应不同类型的多模态数据输入。

为了保证特征提取的准确性，需要在特征提取层进行各个提取特征的对齐工作，以确保特征提取的有效性。

为了提高图文检索的效率，这里将对齐后的特征压缩成二进制哈希码，方便检索时的相似度计算和匹配。同时为了进一步提高图文检索效率，这里将特征压缩后的二进制码进行聚类分析，进行分类，在检索时根据聚类结果缩小图文检索的范围。

1.2　特征提取与对齐

在针对文本－图像信息的特征提取和对齐上，需要设计一个多模态的 Transformer 模型。该模型主要由以下组成部分：

1）编码器：多个编码器将输入的文本和图像特征进行编码，生成对应的文本和图像特征向量。每个编码器由多个层次组成，每层都包括一个多头自注意力模块和全连接前馈网络模块。输入的图像特征由图像编码器通常使用预训练的卷积神经网络（如 ResNet[18]、VGG 等）提取图像特征。这些特征经过降维和归一化处理后，可以作为输入传递给后续的模块。其网络架构如图 2 所示。

2）多头注意力机制：多模态注意力机制允许模型在文本和图像之间进行交互。它通过计算文本和图像特征之间的相似度来确定哪些文本单词对应哪些图像区域。以实现针对图像和文本特征的对齐，这样可以实现文本和图像之间的语义对齐。

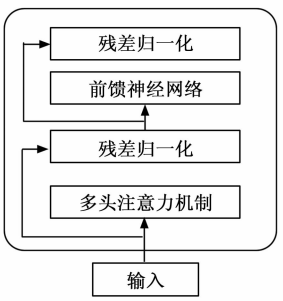

图 2　Transformer 编码器

多模态注意力机制通过计算图片和文本之间的相似度，来实现图像－模态之间的交互与信息传递，从而提高特征表示。假设图片向量为 x，文本向量为 y，首先可以通过计算 x 和 y 之间的相似度矩阵 $\boldsymbol{M}$ 来描述它们之间的关系。具体来说，$\boldsymbol{M}_{i,j}$ 表示 x_i 和 y_i 之间的相似度，使用余弦相似度等方法进行计算。

$$\boldsymbol{M}_{i,j} = f(x_i, y_i) \tag{1}$$

其中：$f(\cdot)$ 是余弦相似度函数，它衡量了图片向量和文本向量之间的相似程度。对于任意两个相同维度的实数矢量 $x_i = (x_1, x_2, \cdots, x_j)$ 和 $y_i = (y_1, y_2, \cdots, y_j)$，它们的余弦距离可以计算如下：

$$d_{\cos}(x_i, y_i) = 1 - \frac{x_i^T y_i}{|x_i| \cdot |y_i|} \tag{2}$$

1.3　聚类

为了提高图文检索的计算效率，这里使用特征压缩，将提取并对齐后的特征进行哈希编码，将高维特征向量映射到低维二进制编码，以便进行快速的相似性搜索。

同时为了缩小图文检索中进行计算比较的范围，这里针对哈希码进行聚类分类，以在图文检索中提高检索速度和效率。通过将相似的哈希码聚集到一起，可以缩小计算范围，减少冗余计算。这样，在进行图文检索时，只需要对每个聚类中的代表性哈希码进行计算，而不需要对所有哈希码逐一比对。这种方式可以大幅提高检索速度，并且保持了较高的准确性。利用先进的聚类算法和分类技术，对大量的哈希码进行聚合分析，从而实现更高效的图文检索。

由于数据量的不断增大，能够不重复表示的哈希码也不断增加，导致了哈希编码的长度较长，常见的跨模态哈希编码有 16、32、64 位。较长的哈希编码包含较多的数据信息和较高的信息密度，但是在进行聚类时可能会面临维度灾难（Curse of Dimensionality）的挑战，即高维空间中的样本稀疏性导致聚类算法效果下降。为了进一步缩小哈希编码的维度，缓解维度灾难和计算复杂度问题，同时尽可能保持数据的重要特征，本文在使用深度聚类方法之前，通过引用自编码网络[19]（Autoencoder）将哈希码进一步压缩到三维空间中，然后通过聚类方法将空间中的位置表示进行聚类，形成多个簇实现哈希编码在自编码空间的分类，进而在跨模态检索过程中，将庞大的数据对比过程减少为针对各个自编码空间分类中的数据对比工作，进而提高跨模态检索的效率。基于聚类的图文检索模型如图 3 所示。

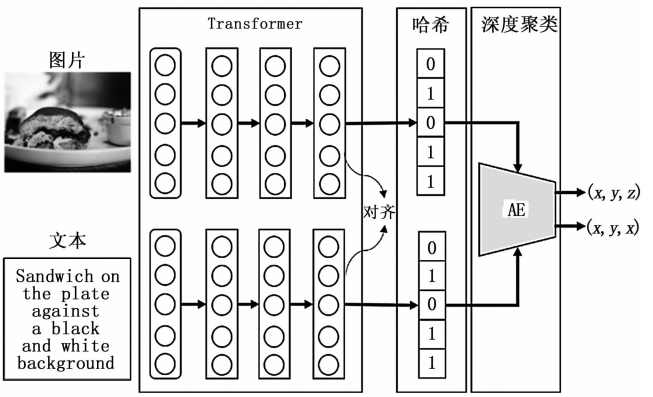

图 3 基于聚类的图文检索模型

需要注意的是，在常见的图文检索中，已经有不依赖哈希编码的子空间的图文检索方法，其内容也是通过将各模态的特征表示降维映射到同一子空间中进行跨模态检索。这里为了进一步提高图文检索效率，可以将 N 位哈希编码的数据视作 N 维数据的点合集（N 为 2 时，取之仅为 {(0，0)，(0，1)，(1，0)，(1，1)}），N 的数值越大，数据表示的精确程度越高，两个数据之间的相似度计算越准确，本方法首先获取数据的较高精确表示，通过深度聚类，将数据映射到低维空间并进行聚类，可以缩小数据的检索范围，当计算数据之间的相似性时，仍然使用哈希编码进行计算，这样可以计算簇内数据的准确相似度，保证了簇内数据检索的精度。

1.4 图文检索流程

本文提出的图文检索模型在针对图像－文本的学习和聚类后，最终的图文检索流程以文本检索图片为例进行介绍。

聚类索引：首先在聚类后，模型将为每个聚类分配一个代表性的簇中心点 O_i（x，y，z）。这些代表性中心点将用于后续的图文检索。聚类后的图片子空间分布如图 4 所示。

查询处理：当进行图文检索时，首先对查询文本 q 进行哈希码生成及聚类编码 c。

范围缩小：确定需要在哪些聚类中进行详细的图文匹配。将查询文本 q 与聚类索引 O_i 进行比较，通过计算查询的聚类编码 c 和聚类索引 O_i 之间的距离，找到与查询文本距离最接近的聚类。只有位于相似聚类中的图片才会被选中，从而缩小了计算范围。将查询文本 q 与聚类索引 O_i 的相似度由闵可夫斯基距离来计量。闵可夫斯基距离用于测量两个实向量之间的相似性，可以通过 L_p 范数计算。对于任意两个相同维度的实维向量 $\boldsymbol{a}=(a_1, a_2, \cdots, a_j)$ 和 $\boldsymbol{b}=(b_1, b_2, \cdots, b_j)$，它们之间的闵可夫斯基距离为：

$$L_P(\boldsymbol{a},\boldsymbol{b}) = \left(\sum_{i=1}^{j} | a_i - b_i |^P\right)^{1/P} \tag{3}$$

为了进一步明确图文匹配的范围，我们这里设置一个最大阈值 $L_{\max}$，当计算查询的聚类编码 c 和聚类索引 O_i 之间的聚类小于 $L_{\max}$ 时，则选取距离簇中心最小的簇进行图文匹配检索，当计算查询的聚类编码 c 与所有的聚类索引 O_i 的距离均大于 $L_{\max}$，则选取距离最小的两个簇进行图文匹配检索。

图文匹配：在经过范围缩小后，对选定的聚类中的哈希码进行更详细的图文匹配。使用进行哈希编码后的图像二进制表示和查询文本的二进制表示，计算它们之间的汉明距离。

汉明距离是指不同等长字符串中的字符数。对于任意两个长度相同的二进制向量 $\boldsymbol{a}$，$\boldsymbol{b}\in(0, 1)^j$，它们之间的汉明距离可以通过异或运算来计算，即：

$$d_h(\boldsymbol{a},\boldsymbol{b}) = \sum_{i=1}^{j} xor(a_i, b_i) \tag{4}$$

结果返回：根据图文匹配的结果，返回与查询文本最相似的内容作为检索结果。

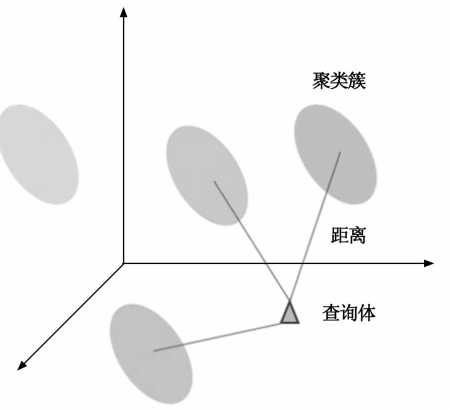

图 4 图片聚类示意图

2 哈希编码的深度层次聚类

基于哈希码的深度聚类方法首先使用自编码器（Autoencoder）学习哈希码的子空间特征，将多位哈希码压缩到一定范围的子空间之中，针对哈希码的子空间分布应用聚类算法，将子空间分布进行分类，以减少检索的范围。自编码器的训练框架如图 5 所示。

自编码器由两部分组成：编码器（Encoder）和解码器（Decoder）。编码器将输入数据映射到潜在空间中的低维表示，而解码器则将该低维表示映射到原始数据空间。根据

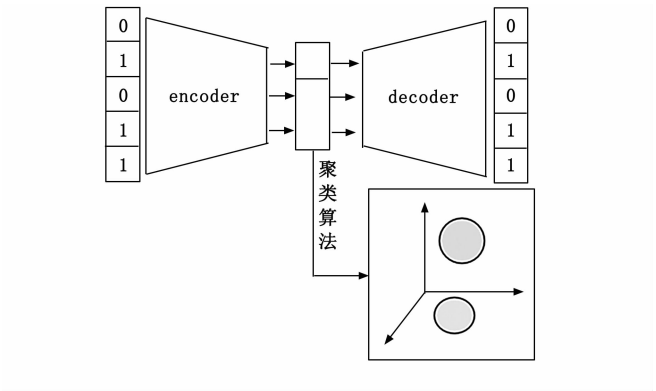

图 5　深度聚类自编码器

任务需求，可以选择不同类型的神经网络模型作为编码器和解码器，如全连接神经网络、卷积神经网络等。由于哈希码的结构简单，维度相对单一，因此编码器和解码器均使用全连接神经网络对哈希码进行学习。在本文的深度聚类任务中，需要对针对编码器生成的子空间表示应用聚类算法，并将聚类算法的损失计算到自编码器的损失中。

通过自编码器学习到的子空间表示在应用聚类算法后，可以将各相似数据聚集在同一个簇中，而在检索过程中，需要将检索的向量跟簇中的数据进行比较匹配，但是由于聚类算法并未提供准确的标志结果，导致检索的数据无法进行匹配比较。同时，检索的数据由于不在聚类的数据中，可能出现检索的数据在多个簇之间，此时检索的内容不能仅限于单个簇之中。因此，本文任务需要选取的聚类算法需要满足以下条件：1）聚类算法的结果需要提供一个具体的簇信息，便于检索数据对聚类结果进行比较；2）聚类算法的结果需要满足各个簇之间具有一定的距离差异，确保聚类算法可以实现数据的分类，减少数据检索的数量。

通过需求分析和方法匹配，层次聚类算法[20-21]可以满足该任务的需要，层次聚类算法是一种基于距离（或相似度）的无监督学习方法，用于将数据集中的样本分成不同的群组。具体流程如图 6 所示。通过不断聚合相近的簇，不断丰富聚类簇内容，并且不断计算簇的中心信息及聚类簇信息，完成聚类并返回各个簇的中心信息。

因此在损失函数的计算上，将跨模态哈希部分的损失 $Loss_{CT}$，加入深度聚类自编码器的损失 $Loss_{AE}$，即：

$$Loss = Loss_{CT} + Loss_{AE} \tag{5}$$

跨模态哈希模块针对图像哈希码 h_I，文本哈希码 h_t，损失函数为：

$$Loss_{CT} = \sum_{(i,j)\in S} l(h_{I_i}, h_{t_j}) + \sum_{(i,j)\in D} l(h_{I_i}, h_{t_j}) \tag{6}$$

式中，S 是相似图像文本对集合，D 是不相似的图像文本对集合，$l(\cdot,\cdot)$ 是图像和文本哈希码的距离函数，这里选择汉明距离来衡量图像－文本的相似度。

自编码器针对输入的哈希码 H_{code}，进行学习，将复杂的哈希码映射为三维向量 $\boldsymbol{S}_c$，再尝试还原成哈希码 H'_{code}。通过闵可夫斯基距离来衡量自编码器输入前后的哈希码的损失，用以训练自编码器对哈希码的学习能力。

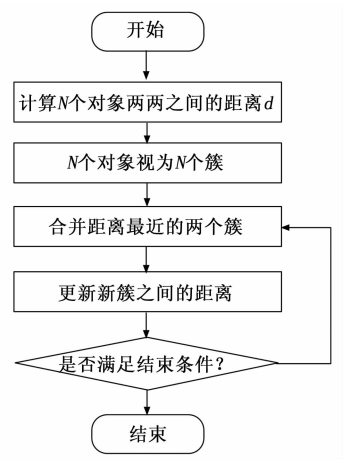

图 6　自底向上的层次聚类流程

$$L_P(H_{code}, H'_{code}) = \left(\sum_{i=1}^{J} | H_{codei} - H'_{codej} |^P\right)^{1/P} \tag{7}$$

对于应用于本文任务的聚类算法来说，最好的结果是将所有哈希码分类为至少两个的簇，并保持簇之间的分离性较大，这样能够保证在不同簇之间哈希码匹配程度都低于簇内的匹配程度，以提高跨模态检索的精度。因此需要对 S_c 进行一系列聚类运算，并通过借鉴轮廓系数[21]（Silhouette Coefficient）来评估聚类的效果。

计算轮廓系数需要以下步骤：

1）对于每个样本 i，计算其与同簇其他样本的平均距离 $a(i)$；

2）对于每个样本 i，计算其与最近簇中所有样本的平均距离 $b(i)$；

3）对于每个样本 i，计算其轮廓系数 $s(i)$，公式如下：

$$s(i) = \frac{b(i) - a(i)}{max[a(i), b(i)]} \tag{8}$$

4）对于所有样本的轮廓系数求平均得到整个聚类结果的轮廓系数。

对于每个样本，轮廓系数的取值范围在－1～1 之间。当轮廓系数接近 1 时，表示样本聚类合适；当轮廓系数接近－1 时，表示样本更应该被分到其他簇；而当轮廓系数接近 0 时，则表示样本处于两个簇的边界上。

为了避免簇之间距离过近导致聚类结果影响跨模态检索的精度，因此需要确保同簇内平均距离较小，而异簇之间平均距离较大。故聚类的轮廓系数趋近 1 最佳，最后结合自编码器输入前后的损失，最终自编码器的损失为：

$$Loss_{AE} = L_P(H_{code}, H'_{code}) - \partial s(i) \tag{9}$$

式中，∂ 代表平衡系数，用于平衡自编码器和轮廓系数在深度聚类损失中的权重。

3　实验结果与分析

3.1　实验设置

本文提出的模型主要在 Linux 系统上构建，在英伟达 3090 显卡上进行训练，模型架构通过 Pytorch 框架进行搭建，具体的开发环境为 Jupyter Notebook，Python 版本为 3.9.4。

实验分别从检索速度和检索精度方面进行。与跨模态哈希检索模型 DCHM[13] 和 SSAH[14] 相比较。基于检索速度的实验主要是检验提出的聚类思想可以显著地提高跨模态检索模型的速度，以验证本模型的可用性。基于检索精度的实验主要是检验聚类思想的使用，在实现跨模态快速检索的同时，保证检索精度维持在可靠的范围内，用以验证本模型的可靠性。

3.2 数据集及评价标准

COCO[22] 和 Flickr30K[23] 是两个常用的跨模态检索数据集，用于训练和评估跨模态检索任务的模型。COCO 字幕包含 123 287 张来自 MSCOCO（Microsoft Common Objects in Context）数据集的图像，以及每张图像人为生成的 5 个标题。除去不常见的单词后，标题的平均长度为 8.7。数据集被分成 82 783 张训练图像，5 000 张验证图像和 5 000 个测试图像。Flickr30K 是一个包含约 31 000 张图像的跨模态数据集，每张图像都伴随着 5 个句子级别的描述。该数据集收集了来自 Flickr 图像分享网站的图像，并由人工撰写了多个描述。图像内容涵盖了各种场景、对象和活动，适用于多样化的跨模态检索任务。每个图像包含 5 个文本描述。数据集分为 3 部分，1 000 张图片用于验证，1 000 张图片用于测试，其余的用于训练。

mAP（平均精度均值，Mean Average Precision）是一种常用的评估指标，常用于信息检索领域。它用于衡量在排序任务中，模型对相关文档的排序质量。在信息检索中，常常需要根据用户查询，从一个文档集合中对文档进行排序，以便将最相关的文档排在前面。mAP 通过计算检索结果中每个查询的平均精度 AP（Average Precision），再对所有查询的平均值进行计算，来评估模型在整个数据集上的性能。mAP 的计算公式如下：

$$mAP = \frac{(AP_1 + AP_2 + \cdots + AP_n)}{n} \tag{10}$$

其中：n 表示查询的总数，AP_i 表示第 i 个查询的平均精度。对于每个查询 i，计算其平均精度（AP_i）的步骤如下：对于每个相关文档 j，计算其在检索结果列表中的位置 rank（j）；计算精度（Precision），即在 rank（j）之前的所有文档中，与查询相关的文档数量除以 rank（j）；将所有相关文档的精度求和，并除以相关文档的总数 n，得到平均精度（AP_i）。

为了量化比较检索速度的差异，需要计算平均查询速度 AT，即计算全部查询完成的时间总和，并除以查询的总数 n，计算公式如下：

$$AT = \frac{(T_1 + T_2 + \cdots + T_n)}{n} \tag{11}$$

4 实验结果及分析

本文实验由两部分构成。首先在公开数据集上进行检索速度和检索精度的实验比较，以验证本文提出模型的有效性和优越性。其次，为了验证层次聚类算法对本文提出模型的影响，取消自编码模块并采用不同的聚类算法进行消融实验。

4.1 模型有效性实验

在跨模态检索实验中，通常按照图像检索文本（IQT，image query text）和文本检索图像（TQI，text query image）进行对比分析。

首先进行跨模态检索效率的比较，计算跨模态检索的平均速度（单位为秒），结果如表 1、表 2 所示。

表 1 在 MSCOCO 数据集上跨模态检索时间比较

任务	方法	编码长度		
		16 位	32 位	64 位
IQT	DCHM	1.64	1.64	1.62
	SSAH	1.34	1.34	1.33
	本文方法	1.12	1.11	1.07
TQI	DCHM	1.47	1.52	1.50
	SSAH	1.21	1.23	1.30
	本文方法	1.03	1.02	1.03

表 2 在 Flickr30K 数据集上跨模态检索时间比较

任务	方法	编码长度		
		16 位	32 位	64 位
IQT	DCHM	1.42	1.43	1.39
	SSAH	1.23	1.22	1.19
	本文方法	1.13	1.15	1.13
TQI	DCHM	1.33	1.42	1.30
	SSAH	1.14	1.05	1.01
	本文方法	1.03	0.98	1.06

由表 1、表 2 可知，跨模态检索的效率与哈希编码的长度关系不大，哈希编码的长度主要是提高哈希码对各模态数据表示的精度，长度越长，所表示数据精度越高，最终展现在跨模态检索的精度越高。

本文提出的方法相比 DCHM[12] 方法，无论 IQT 还是 TQI，都显著提升了跨模态检索的平均效率，检索时间在两个数据集上平均提高了近 0.4 s；而对于 SSAH[14] 模型，其将相似度较高的数据归置与同一个桶中进行分类的跨模态检索，跟本文提出的方法类似，都是基于减少跨模态检索范围以提高效率，但是本文提出的方法可以更准确地对被检索数据进行分类，因此检索时间在两个数据集上平均提高 0.15 s。

在明显提高检索效率的基础上，进行跨模态检索精度的比较，使用 mAP 平均精度来衡量各个方法的效果，在 MSCOCO 和 Flickr30K 两个数据集上进行跨模态检索精度实验，分别在哈希编码长度设置为 16、32 和 64 位条件下进行跨模态检索精度比较，实验结果如表 3 和表 4 所示。

通过表 3 和表 4 中可知，本文提出的方法相比所选取的两种基准，平均精度介于 DCHM 和 SSAH 之间，相比 DCHM 模型，本文提出的方法全面提高了图文检索的速度和精度。相比 SSAH 模型，本文提出的方法在保证检索平均精度的稳定的情况下，检索效率有一定的提高。实验证

明本文提出的方法可以显著提高跨模态检索的效率，同时保证了检索的精度，具有较好的研究结果。

表 3　在 MSCOCO 数据集上跨模态检索平均精度

任务	方法	编码长度		
		16 位	32 位	64 位
IQT	DCHM	0.511	0.513	0.527
	SSAH	0.550	0.558	0.557
	本文方法	0.533	0.540	0.544
TQI	DCHM	0.501	0.503	0.505
	SSAH	0.537	0.538	0.529
	本文方法	0.502	0.511	0.509

表 4　在 Flickr30K 数据集上跨模态检索平均精度

任务	方法	编码长度		
		16 位	32 位	64 位
IQT	DCHM	0.735	0.737	0.750
	SSAH	0.782	0.790	0.800
	本文方法	0.755	0.762	0.779
TQI	DCHM	0.763	0.764	0.775
	SSAH	0.791	0.795	0.803
	本文方法	0.751	0.780	0.786

4.2　消融实验

本文提出的方法和模型的关键是自编码网络与聚类算法的选取，因此消融实验将通过两方面进行，并且主要对比跨模态检索的效率，这样更能体现本文方法的可行性和有效性。

首先，对自编码器进行拆解，去掉解码器部分，只保留编码器和聚类算法模块。在本文提出模型上使用 MSCOCO 数据集进行跨模态检索时间的比对。由图 7 的结果可知，在去掉解码器模块后，跨模态检索的时间大幅度地提高。其原因在于去掉解码器模块后自编码器失去了重构输入数据的能力。从本文提出的方法而言，自编码器的目标是学习数据的有用表示，以便于压缩和重构。解码器帮助自编码器学习如何在编码过程中保留输入数据的关键信息。如果没有解码器，可能会导致编码器学习到的表示不够有意义或有用。

其次，通过更换聚类算法，选取 K-means，DBSCAN（DBSCAN，density-based spatial clustering of applications with noise）等常见的聚类算法，对本文使用的层次聚类算法进行替换。并在本文提出的模型上使用 MSCOCO 数据集上分别对比不同编码长度跨模态检索速度。通过图 8 中多种聚类算法的比较，可以得出层次聚类在跨模态哈希检索的效率提升方面有着良好的表现。在聚类结果的检索速度上，层次聚类算法通常具有一定的优势。这是因为层次聚类算法生成了一个树状结构，可以根据需要在树上进行不同层次的切分，从而实现对聚类结果的快速检索。例如，可以根据特定的标准（如距离）在树状结构中进行剪枝，

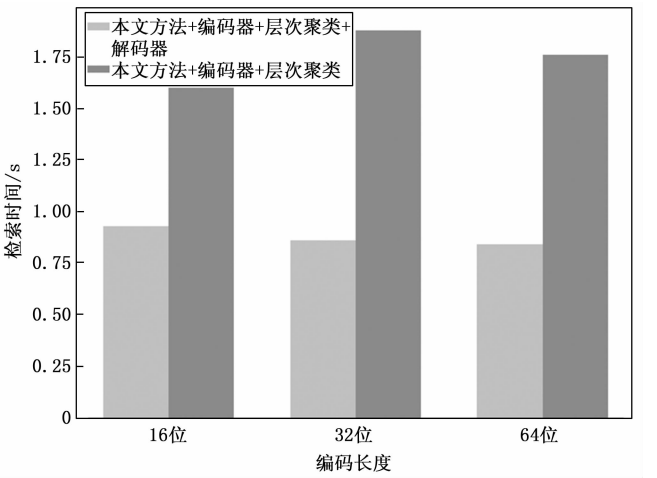

图 7　消融实验

以加速聚类结果的检索过程。相比之下，K-means 和 DBSCAN 在获得聚类结果后，需要额外的步骤才能实现类似的检索速度优势。

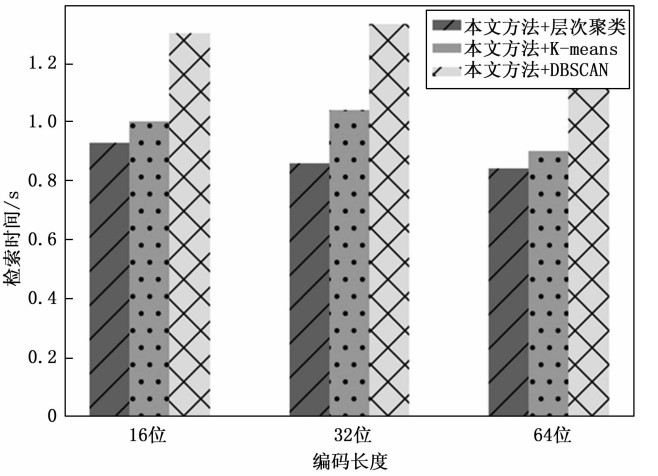

图 8　不同聚类算法比较

5　结束语

本文主要介绍了一种基于聚类的跨模态快速检索技术。该技术针对哈希码进行深度聚类。在跨模态检索过程中，只在对应的聚类簇中进行检索，以提高检索速度。通过实验，证明了该方法在保证检索精度的同时显著提高检索速度。本文所提出的方法聚焦于已有模型的进一步改进和性能提升，但是缺少针对跨模态匹配、跨模态表示学习、哈希码生成的研究，这导致本文提出的方法在应用中不能全面地提升跨模态哈希检索的性能。在未来的研究中，将着重于分析跨模态匹配、哈希码生成，更加全面快速地实现跨模态哈希检索。

参考文献：

[1] 张飞飞，马泽伟，周　玲，等. 图文跨模态检索研究进展[J]. 数据采集与处理，2023，38（3）：479－505.

（下转第 298 页）