第 2 章 基于哈希的图像检索相关技术

2.1 引言

基于哈希的图像检索算法是当前图像检索领域研究的热点。1998年Piotr Indyk 等人首次提出的位置敏感哈希算法将哈希用于近似最近邻搜索，解决了高维数据在线性搜索时产生的“维度灾难”问题。随着时代的进步和新技术的发展，基于哈希的图像检索技术也在不断完善和进步。本章将介绍基于哈希的图像检索相关技术，包括算法流程，相似性度量，优缺点和目前存在的问题。

2.2 基于哈希的图像检索基本框架

基于哈希的图像检索算法因为其占用空间小，检索速度快的优点成了图像检索领域研究最广泛的算法。一般来说，基于哈希的图像检索算法流程如图2-1所示。首先，对训练数据集进行特征提取，提取图像的高维特征，然后通过设计好的算法对这些特征进行哈希函数学习，得到学习的哈希函数。将这些高维特征经过学习到的哈希函数映射之后得到哈希码存入数据库。当要进行图片检索时，同样对查询图片进行特征提取再进过哈希函数，得到对应的哈希码。将这个哈希码与数据库中的哈希码一一比对，找到相似的样本，并返回给用户。它主要可以分为以下三个部分：

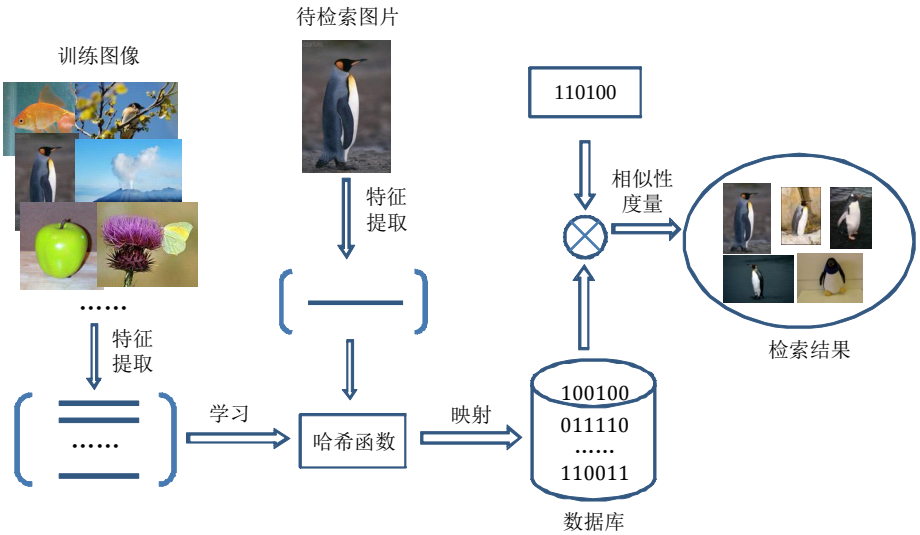

图 2-1 基于哈希的图像检索过程

（1）图像特征提取：特征提取是对图像数据进行分析，提取图像本身内容的向量表示或描述，得到的这些表示向量就被称为图像特征，而提取的这个过程被称为特征提取过程。有了这些图像的向量表示就可以通过训练的方式教计算机去学习如何识别这些图像。图像表示向量的维度往往很高，因为图像的结构和本身内容比较复杂。图像的高维特征使得图像检索过程变成了在高维特征空间中的最近邻检索，但是大规模数据和高维空间下，进行线性的最近邻检索效率非常低，无法满足图像检索系统的需求。

（2）哈希函数学习：为了解决大规模图像检索时存在的“维度灾难”问题，提出了哈希的方法。哈希函数学习的目的是找到一个合适的哈希函数，将高维图像特征映射到低维汉明空间中，同时保证尽量保留原始样本的近邻关系和相似性，即在原始特征空间中相似的样本，在汉明空间也相似。通过对这些低维的哈希码进行索引来进行图像检索，能有效提高图像检索的速度和精度。所以哈希学习这一步最主要的研究内容就是如何学习一个最优的哈希函数，提高检索的正确率。

（3）相似性度量：相似性度量是对不同图像之间相似性大小度量，用于对检索结果的排序，采用不同的相似性度量方法往往会得到不同的检索结果。目前常用的相似性度量标准有欧式距离、汉明距离等。

基于哈希的图像检索不需要存储原始高维图像特征，只需要存储低维度二值哈希编码，可以大大减少存储空间。而且在检索时采用哈希码可以大大加快检索速度。

2.3 图像特征提取

图像特征提取是图像检索的第一步，特征提取的好坏，决定了图像检索结果的好坏。图像特征分为传统的手工特征和深度学习特征。通常传统的手工图像特征类型有颜色、纹理、形状以及局部描述子。

颜色特征是图像特征中一种非常直观的特征，常用的颜色特征有颜色直方图和颜色相关图。颜色直方图是对图像中每种颜色分布的统计量，它不考虑像素的位置，所以对于图像的形变和旋转鲁棒。颜色相关图不仅考虑颜色信息，也考虑了位置信息，它同时根据颜色和位置来构建直方图。

图像纹理特征是一种被广泛使用的全局性特征，它描绘的是图像中物体的表观性质，即在物体表面呈现周期性变化的排列组织属性，它能够区分具有相似颜色特征的图片。纹理特征被广泛应用于基于内容的图像检索系统中。

形状特征常用于检索特定目标对象，其含有一定的语义信息，形状特征

描述的越准确，图像检索的结果越好。一个好的形状特征应具有完备性、独特性、抽象性和几何不变性。形状特征一般有两种，一种是基于图像边缘轮廓信息的形状特征，另一种是基于图像局部信息描绘的形状特征。

局部描述子是目前图像检索领域中最常用的图像特征，比如SIFT[29]、SURF[30]、ORB[31]等，在这些局部描述子的基础上，用不同的编码方式可以构建出不同的图像的全局描述，具有代表性的编码方式有局部特征聚合描述符（Vector of Locally Aggregated Descriptors, VLAD）[32]、词袋模型（Bag of Words, BoW）[33]和Fisher向量（Fisher Vector，FV）[34]。由于将局部描述子编码成全局的特征描述，且这些局部描述子本身具有尺度、平移和旋转不变性，这些以局部描述子作为特征的图像检索方法一般来说都能够获得比较好的检索效果，但这一类方法通常需要很高的数据表达维度，比如上万维，才能达到比较好的效果。

而随着深度学习的发展，出现了很多用深度神经网络提取的特征，这些深度学习特征是从数据集中自动学习的，而不是采用手工方法进行提取的。在过去的模式识别、机器学习领域中大多数算法都是使用的手工特征，手工特征是根据预先设计的提取规则对图像信息进行提取，生成的对图像的向量表示。深度学习特征相比手工特征，主要有两点优势。第一点是手工特征的好坏依赖于提取规则设计的好坏，针对不同的学习任务，不同的提取规则效果差别很大，且针对具体任务设计出一个有效的手工特征需要大量的先验知识。而深度学习可以从不同的数据集，和不同的学习任务中自动且快速地学习到有效的特征。第二点是使用手工特征进行机器学习任务，特征提取过程和机器学习过程是分开的，特征提取的过程不能够进行学习。而利用深度学习，可以把这两步结合在一个网络中同时训练，提升整体的性能。

2.4 相似性度量

相似性度量作为衡量特征之间的相似度依据，是图像检索过程中重要的一步，采用不同的相似性度量方法往往会得到不同的检索结果。因此，合适的相似性度量方法也非常重要。相似性度量主要分为基于距离的度量方式和基于相似度的度量方式。

2.4.1 基于距离的度量方式

假设x_1、x_2和x_3分别表示三个不同图像的特征向量，让$D(\cdot,\cdot)$表示距离计算

函数，则$D(\cdot,\cdot)$应满足以下四个基本定理：

对称定理：

$$D(x_1, x_2) = D(x_2, x_1) \tag{2-1}$$

自相似定理：

$$D(x_1, x_1) = D(x_2, x_2) = D(x_3, x_3) = 0 \tag{2-2}$$

三角不等式定理：

$$D(x_1, x_2) + D(x_2, x_3) \geq D(x_1, x_3) \tag{2-3}$$

最小定理：

$$D(x_1, x_2) \geq D(x_2, x_3) = 0 \tag{2-4}$$

下面介绍几种常用的距离度量方式：闵可夫斯基距离、汉明距离、马氏距离等。

1) 闵可夫斯基距离

闵可夫斯基距离是定义在p范数空间的距离度量，是最基本的计算实数向量相似度的方式。设两个d维二值向量u和v，则L_p距离定义为：

$$L_p(u, v) = \left[\sum_{i=1}^{d} |u_i - v_i|^{\mathrm{p}}\right]^{\frac{1}{p}} \tag{2-5}$$

其中u_i，v_i分别表示向量中第i维比特位。闵可夫斯基距离越小，它们越相似。闵可夫斯基距离在p取不同值时，有不同的结果。

当$p = 1$时，式（2-5）变成$L_1(u, v) = \sum_{i=1}^{d} |u_i - v_i|$，此时的距离度量就变成常见的曼哈顿距离。

当$p = 2$时，式（2-5）变成$L_2(u, v) = \sqrt{\sum_{i=1}^{d} |u_i - v_i|^2}$，此时的距离度量转化为常用的欧式距离。

当$p = \infty$时，式（2-5）变成$L_\infty(u, v) = \max_{i=1}^{d} (|u_i - v_i|)$，此时的距离变成了两个向量最大差距维度的差，被称为切比雪夫距离。

2) 汉明距离

汉明距离主要是用来计算两个离散型向量之间的相似性。假设有两个d维离散向量u和v，那么它们之间的汉明距离定义为同比特位上编码不同的比特位数目：

$$H(u, v) = \sum\nolimits_{i=1}^{d} (u_i \neq v_i) \tag{2-6}$$

其中u_i，v_i分别表示向量中第i维比特位。两个向量之间的相似度随着汉明距离

的增大而减小。计算机计算汉明距离的速度非常快，因为这只需要少量的二进制位运算。

3) 马氏距离

马氏距离是一种尺度无关的距离度量单位，它消除了不同维度之间的相关性，和不同维度之间的尺度差异。马氏距离的表达式如下：

$$D_{mahat}(u,v) = (u-v)^T\Sigma^{-1}(u-v) \tag{2-7}$$

其中Σ是u,v之间的协方差矩阵。同样的，两个向量之间的相似度随着马氏距离的增大而减小。

2.4.2 基于相似度的度量方式

相似度通过使用相似度函数来测量两个特征向量之间的相似程度。取不同的相似度函数所算出的相似度是不同的。相似度函数值越大，两个特征向量之间的相似度越大。基于相似度的度量方式主要有两种，夹角余弦和Jaccard相似度。

1) 夹角余弦相似度

夹角余弦相似度首先计算两个向量的夹角余弦值，用两个向量的夹角余弦值去度量两个向量差异大小。夹角余弦值的取值范围为$[-1,+1]$。两个向量之间的相似度随着夹角余弦的增大而增大。设有两个实数向量u,v，它们的夹角余弦相似度定义为：

$$\cos(u,v) = \frac{u\cdot v}{|u||v|} \tag{2-8}$$

2) Jaccard相似度

Jaccard相似度常用于比较样本集之间的差异性，即集合之间的相似度。定义为两个集合的交集大小除以并集的大小。设有两个集合X和Y，它们的Jaccard相似度为：

$$J(X,Y) = \frac{|X\bigcap Y|}{|X\bigcup Y|} \tag{2-9}$$

2.5 传统图像哈希技术

本节介绍几种传统的图像哈希方法，包括位置敏感哈希方法（Locality sensitive hashing，LSH）、谱哈希方法（Spectral hashing, SH）、核监督哈希算法（Supervised Hashing with Kernels, KSH）。这些方法都是经典的基于手工特征的传统哈希算法。

2.5.1 位置敏感哈希方法

最早用哈希来做图像检索的算法是位置敏感哈希算法[5]，它实现了图像特征的近似最近邻检索。LSH背后的关键思想是将图片特征数据通过哈希函数映射为哈希码，由于哈希码很短，汉明距离计算量小。在计算图片特征之间的相似度时，根据其对应的哈希码来进行计算，加快了计算速度。这是一种快速的近似最近邻检索方式。位置敏感哈希算法主要是认为在原始空间中数据点之间距离越小，碰撞的概率越大。设数据集$X = \{x_i\}_{i=1}^{N}$，$D(\cdot,\cdot)$是数据集中两个点的距离度量函数，$x_p \in X$ 和$x_q \in X$是数据集中任意两个样本，$h(\cdot)$是哈希函数，则：

如果$x_p \in C(x_q, r_1)$，则有$\Pr_H[h(x_q) = h(x_p)] \geq p_1$

如果$x_p \notin C(x_q, r_2)$，则有$\Pr_H[h(x_q) = h(x_p)] \leq p_2$

其中$C(x_q, r)$是以x_q 为圆心，r为半径构成的一个圆形闭合区域，p_1和p_2是两个不同的概率值，r_1 和r_2 是检索半径，满足这样条件的哈希函数称为(r_1, r_2, p_1, p_2)敏感的哈希函数。一般来说，$r_1 < r_2$且$p_1 > p_2$。

图2-2是位置敏感哈希算法的编码流程。算法随机的产生k个超平面将高维特征样本进行划分，当样本位于超平面的同一侧赋予相同的比特位，不同侧赋予不同的比特位，最后得到每个数据点对应的k比特哈希码值。

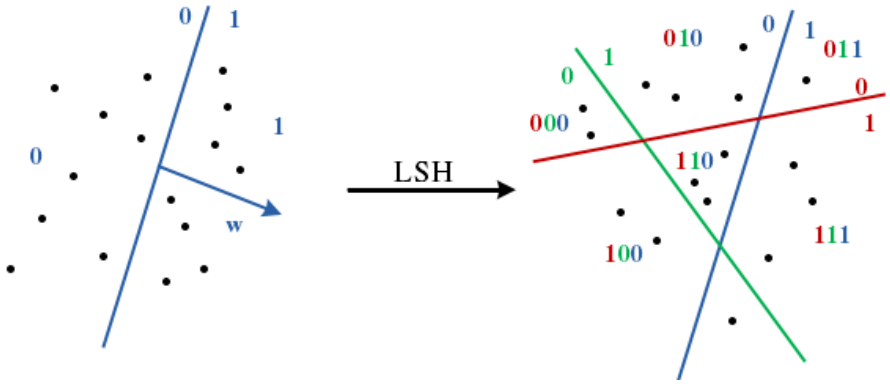

图 2-2 位置敏感哈希算法的编码流程

LSH算法是一种非数据依赖的哈希方法，它的哈希函数是随机产生的，并不依赖于训练集。这些哈希函数的形式为：

$$h(x) = \mathrm{s}\,ign(W^T x + b) \tag{2-10}$$

其中，$sign(\cdot)$是符号函数，W是投影权值矩阵，x是原始高维数据，b是可变的偏移量。但是，为了获得好的性能，LSH方法要求很长的比特位，这会大大减少检索的召回率，而且由于LSH是随机产生的哈希投影，并没有根据数据集

去学习，使得它无法获得原始高维特征的分布结构，因此，该算法还存在很多可以改进的地方。此后，出现了很多对LSH算法进行改进的算法，如p-stable LSH、KLSH、SIKH等等，这些算法在一些方面提升了LSH的性能，但是他们还都是非数据依赖的算法，并不能很好的利用数据本身的信息。所以之后数据依赖的哈希算法，即哈希学习算法，成了基于哈希的图像检索算法中研究的主流。

2.5.2 谱哈希方法

谱哈希[13]是2008年在NIPS上发表的算法，其在语义哈希算法的基础上引入谱分析技术，使得映射后的哈希码在汉明空间中的距离与原始空间中的距离一致。

在谱哈希的目标函数中，期望输入空间中的相似样本在汉明空间中也相似，并要求哈希码-1，+1码值数量平衡且不相关。因此可得谱哈希的损失函数为:

$$\begin{aligned} &\min \sum_{ij} \tfrac{1}{2} A_{ij} \| y_i - y_j \|^2 \\ &s.t.\ y_i \in \{-1, +1\}^K \\ &\sum_i y_i = 0 \\ &\tfrac{1}{n} \sum_i y_i y_i^T = \mathrm{I} \end{aligned} \tag{2-11}$$

其中$A = \{A_{ij}\}_{i,j=1}^{N}$是训练数据的相似度矩阵，根据核函数定义为$A_{ij} = \exp(- \| x_i - x_j \|^2 / \varepsilon^2)$，其中参数$\varepsilon$定义了与相似项对应的欧式空间的距离。平衡约束$\sum_i y_i = 0$保证了每一个哈希比特位能够编码均衡。而正交约束$\frac{1}{n} \sum_i y_i y_i^T = \mathrm{I}$在哈希比特位之间施加正交性以尽量减少冗余。

直接求解上述优化问题是几乎不可能的，因为这个问题本质上是一个NP难的图分割问题，无法在线性时间内求解。同时平衡约束和正交约束使上述问题求解更加困难。受到谱图分析的启发，谱哈希算法先对输入数据进行谱图分析，根据谱图分析结果引入松弛条件，将NP难的原始问题转化成拉普拉斯图问题从而求解出哈希码。

具体的，谱哈希算法通过引入拉普拉斯矩阵L，并放松y_i只能取-1和+1的限制条件，即y_i可以为实数。求解可以采用流形学习[35]的方法，求拉普拉斯矩阵的特征向量，然后对其进行阈值化得到哈希码:

$$\begin{aligned} &\min \tfrac{1}{2} tr(Y^T L Y) \\ &s.t. Y^T 1 = 0 \\ &\tfrac{1}{n} Y^T Y = \mathrm{I} \end{aligned} \tag{2-12}$$

其中Y是一个实数的N×K的矩阵，它的第j行是y_j^T，$L = D - A$，D是一个对角矩阵，$D(i,i) = \sum_j A_{ij}$。在求解得到Y之后，再s $ign(Y)$得到对应的哈希码。

在对测试数据集计算哈希码时，首先使用主成分分析算法（Principal Component Analysis, PCA）[36]对测试数据进行降维，然后对降维后的数据矩阵计算其特征向量，最后将特征向量进行阈值化处理，得到最终的哈希码。算法的关键点是要对原始离散约束进行松弛，并将原始问题转化为拉普拉斯矩阵求特征向量的问题。最后得到的解是原始问题的近似值。

谱哈希算法是一种数据依赖的哈希方法，它利用数据本身的分布信息去学习对应的哈希函数，同时它也是一种无监督的哈希方法，它并没有用到数据的标签信息。所以相对于非数据依赖的方法，它的效果要提升很多，同时它也存在很多缺陷。第一是其对于数据分布的假设要求太严格，它要求原始数据服从多维均匀分布，但是实际情况中数据可能并不满足某种特定的分布，它们很有可能处于无序状态。第二它没有使用到标签信息，它仅仅学习到了数据的结构相似性，而学不到数据之间的语义相似性。所以其性能还能够进一步的提高，同时也由于它不需要标签信息，所以该算法可以用于无标注数据。

2.5.3 核监督哈希

无监督哈希方法对于数据集的要求不高，它并不需要数据的标签信息，而监督哈希充分挖掘数据的标签信息，利用标签信息促进生成更好的哈希码。

核监督哈希[22]是哥伦比亚大学刘威教授等人在2012年CVPR上提出的一种有监督哈希方法。该算法利用汉明码内积和汉明距离的等价性，每次顺序且有效的训练哈希函数的一个比特位，从而产生很短的且区分性很强的哈希码。图2-3是核监督哈希的算法流程图。

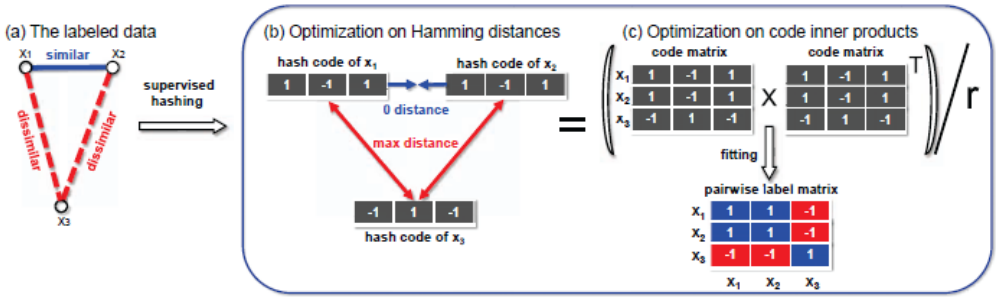

图 2-3 核监督哈希算法[22]

该算法的具体步骤是：

（1）首先定义需要学习的哈希函数：

$$h(x)=\sum_{j=1}^{m}k(x_j,x)a_j-b \tag{2-13}$$

其中，$k(\cdot,\cdot)$ 代表核函数，$x_1,...,x_m$ 是从训练集中均匀选取的m个样本，其主要目的是将原始数据映射到核空间中，使其线性可分。$a_j\in R$ 是核系数，$b\in R$ 是偏差量。

（2）根据训练集的标签信息建立相似矩阵。将类别的相同的图像对标记为1，不同类别的为-1，类别未知的记为0，可以得到如下相似矩阵：

$$S_{ij}=\begin{cases}1,(x_i,x_j)\in M\\-1,(x_i,x_j)\in C\\0,\end{cases} \tag{2-14}$$

其中M是相似样本集，C是不相似样本集。将该相似矩阵作为学习目标，指导哈希函数的学习。

（3）构建目标函数。核监督哈希的目标是使得学到的样本哈希码尽可能符合样本的标签信息分布情况。其目标函数为：

$$\min_{H\in\{-1,1\}^{l\times r}}Q=\left\|\frac{1}{r}HH^T-S\right\|_F^2 \tag{2-15}$$

其中

$$H=\begin{bmatrix}code_r(x_1)\\.....\\code_r(x_l)\end{bmatrix} \tag{2-16}$$

表示要学习的哈希码，r 是哈希码长度，S是相似矩阵，将前面的哈希函数表达形式代入可以得到：

$$H=\begin{bmatrix}h_1(x_1),...,h_r(x_1)\\.....\\h_1(x_l),...,h_r(x_l)\end{bmatrix}=sign(\overline{K_l}A) \tag{2-17}$$

则公式（2-15）可以改写为：

$$\min_{A\in R^{m\times r}}Q(A)=\left\|\frac{1}{r}sign(\overline{K_l}A)sign(\overline{K_l}A)^T-S\right\|_F^2 \tag{2-18}$$

其中A 是学习目标，$\overline{K_l}$ 是核函数。该目标函数利用哈希码与相似矩阵的性质来对哈希码进行优化，使得同类别的图像会尽量映射为相同的哈希码，而不同类别图像会尽量映射为不同的编码。由于$sign(\cdot)$是个非连续函数，所以核监督哈希提出两种松弛方法，sigmoid平滑化和谱松弛方法来对该目标函数进行求

解。

核化的有监督哈希算法充分挖掘数据的标签信息，利用相似矩阵指导哈希码的生成，大大提升了哈希吗的检索性能。但是其还是存在很多不足。首先其使用的是手工提取的高维特征，这种方法在提取时并没有考虑到数据语义信息，且其特征提取这一步和哈希函数学习这一步是割裂开来的，并不能学习到最优的哈希函数。

2.6 深度哈希技术

2012年，Alexnet的提出，奠定了深度卷积神经网络在图像分类任务上无可匹敌的地位。相比传统的手工特征，如颜色，纹理，形状和各种局部描述子，深度卷积神经网络提取的往往具有更强的表示力。它能针对不同的数据集去学习数据内部的表示，在不同数据集上得到的特征都不相同，这样的适应力使得深度卷积神经网络可以适用于很多不同的任务。将深度卷积神经网络用于图像检索任务的研究，可以分为两个阶段。第一个阶段是采用非端到端的形式，即深度卷积神经网络只用于提取图像特征，而哈希函数学习的方式还是用传统的哈希算法。这是最简单的将深度卷积神经网络用于图像检索的方法，提取到的深度特征具有较强的语义表达能力，比人工特征在图像检索任务中取得的效果要好很多。但是，这样将特征提取与哈希函数学习过程割裂开来，不利于我们学习的到最优的哈希函数，而且哈希函数学习优化的结果也不能反作用于特征提取过程，所以就出现了端到端的深度哈希网络。第二个阶段是端到端的网络形式，在原始网络特征提取层之后加入全连接层进行哈希函数学习，通过不断的反向传播优化反馈来提升整体的性能。相比于非端到端的网络结构，这种方法往往能取得更好的效果。

2.6.1 卷积神经网络哈希

最早将深度卷积神经网络与哈希学习结合的算法是卷积神经网络哈希算法（CNNH, Convolutional Neural Network Hashing）[27]，该算法在2014年人工智能顶级会议AAAI上由颜水成提出。它是一种非端到端的方法，能够自动学习图像特征和哈希函数。该算法流程如图2-4所示，总共包含两个阶段。第一阶段是要根据标签信息学习近似的哈希码。首先是设定一个矩阵S表示训练图像对之间的相似度，即如果两张图片拥有相同的类标，记为1，否则记为-1。然后根据目标函数：

$$\min_{H} \left\| S - \frac{1}{q} HH^{T} \right\|_{F}^{2}, s.t. H \in \{-1, 1\}^{n \times q} \tag{2-19}$$

将S分解成HH^T的乘积，其中H是一个矩阵，存储的是训练数据集所对应的近似哈希码。在对H进行求解时采用可缩放的坐标下降法，得到最终需要的近似哈希码。在第二阶段，通过针对H中学习的近似哈希码和可选的图像的离散类别标签，设计一个深度卷积神经网络结构去学习哈希函数。该算法在性能上优于几种最先进的传统监督和无监督哈希方法。

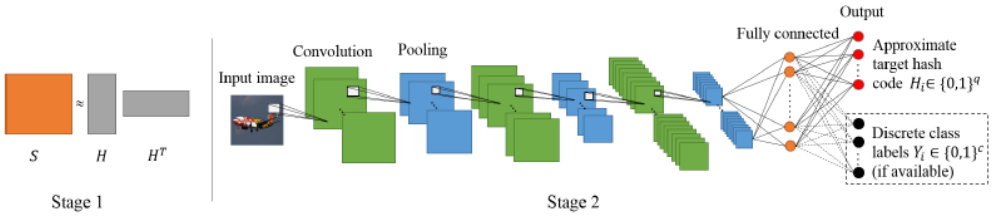

图 2-4 CNNH模型结构[27]

CNNH算法在性能上相比之前传统的基于手工特征的方法提升很大，但是这个方法还是存在不足。首先是这个方法并不是端到端的，学习到的哈希函数并不能反作用于训练集哈希码的更新；其次是其没有很好的考虑量化误差，导致生成的哈希码不能很好的保留原始图像的信息。

2.6.2 深度神经网络哈希

2015年出现了多篇深度学习和哈希学习相结合的文章，其中比较出色的一篇文章是深度神经网络哈希算法（DNNH, Deep Neural Network Hashing）[28]，采用了端到端的模型。它将特征提取与哈希函数学习融合在一个网络中进行学习。

DNNH是来自中山大学的潘炎老师研究组。因为这篇文章采用一个层数很深的卷积神经网络模型，所以简称DNNH，DNNH的模型结构如图2-5。网络的输入是图像的三元组，三元组是指挑选的三张图片其中两张是相似的，即属同一个类，而剩下的那一张和前面两张图片不相似，即不属于同一个类。DNNH算法的目标函数是：对于一个三元组内的三张图片，在经过哈希映射到汉明空间中时，不相似的图片之间的距离要远大于相似图片之间的距离。而且DNNH算法对网络结构做了很多改进，使得该网络能够同时进行特征提取和哈希函数学习，这些改进包括：

（1）为了减少学习到的哈希码不同比特位之间的冗余，DNNH将全连接层用部分连接层代替。每个部分连接学习哈希码的一个比特位，不同比特位之间

没有连接。

（2）为了解决目标函数中哈希码值的离散约束，DNNH采用sigmoid松弛策略，用siggmoid函数将离散约束松弛成连续约束。

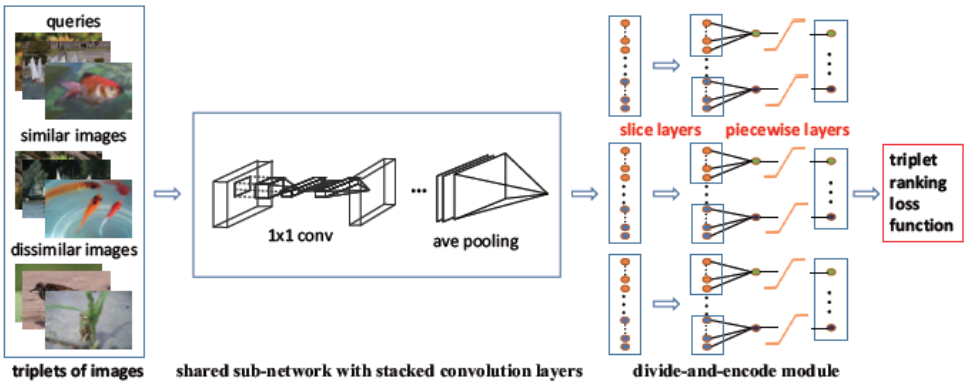

图 2-5 DNNH模型结构[28]

将上面两个改进点结合，就构成了DNNH的divide-encode结构。借助divide-encode模块，DNNH能够对数据集进行端到端的训练，其学习到的哈希码可以反作用于图像特征学习，而学习到的图像特征又能改进哈希码，所以DNNH算法相比于CNNH，图像检索的性能有所提升。

但是，该端到端的模型还是有不足之处。首先是用simoid函数对离散约束的松弛，sigmoid是一种非线性函数，使用这种非线性函数将不可避免地减慢甚至抑制网络的收敛，出现梯度消失的问题。其次是这个网络是采用的三元组标签信息，图像三元组的质量直接受图像三元祖的挑选质量的影响，且图像三元组的挑选需要大量的工作。限制了该算法的应用。

2.7 本章小结

基于哈希的图像检索技术是目前图像检索领域主要的研究方向，它提高了图像检索的精度，同时解决了大规模图像检索中存在的“维度灾难”问题，提高了检索速度。本章详细介绍基于哈希的图像检索的相关内容，包括其基本框架、图像特征提取和相似性度量准则等。之后重点介绍了几种目前比较流行的哈希算法，有位置敏感哈希（LSH）、谱哈希（SH）、核监督哈希（KSH）以及基于深度哈希的算法，包括CNNH和DNNH，分别介绍了这些哈希算法的基本原理，指出他们的优缺点。

第 3 章 基于深度哈希学习的图像检索算法的改进

3.1 引言

目前的深度哈希算法，通过端到端对图像特征提取和二进制码学习，相比传统哈希算法性能有所提升。然而，这些方法仍然存在一些缺点，限制了他们的实际检索性能。首先，相似信息在实际检索系统中通常非常稀疏，即相似图像对的数目远小于不相似图像对的数量。这将导致数据不平衡问题，降低学习到的模型效果。其次，目前大多数深度哈希方法采用sigmoid函数“松弛”的方法来生成连续的近似哈希码，这将导致哈希网络难以收敛，且由于特征经过sigmoid函数之后会损失大量的信息，导致量化误差增大。再次，大多数方法并没有充分利用图像的语义标签信息，图像的语义标签能够提供丰富的语义信息，充分利用语义标签有助于生成语义保留的哈希码，提高系统的检索效率。

本章提出了基于语义保留的深度哈希网络，一种改进的深度哈希学习的网络结构，它主要的创新点是：（1）针对数据不平衡的问题提出了自适应权值的损失函数，减少了数据不平衡带来的影响。（2）针对目前深度哈希学习方法采用sigmoid“松弛”带来的信息损失问题，抛弃了sigmoid“松弛”策略并添加二值约束正则项来减少信息损失和量化误差。（3）为了充分利用图像标签的语义信息，在深度哈希网络后面加入语义保留层，使学习的哈希码保留更多的语义信息。

同时，在语义保留深度哈希的基础上，针对图像数据集有标签数据量少，且获取代价高；而无标签数据量大，易获取却无法被利用的问题。在深度哈希学习中引入生成对抗网络，设计了基于生成对抗网络的半监督哈希学习网络，该网络充分利用无标签数据去提升哈希网络的性能。

3.2 卷积神经网络与哈希学习

卷积神经网络(Convolutional Neural Networks, CNN)是一种典型的神经网络结构，被广泛用于图像识别、目标定位、人脸识别等计算机视觉领域。1998年，Yann LeCun在其论文[37]中首次提出了卷积神经网络，其设计的模型LeNet-5在手写字符识别任务上取得了优异的成绩。CNN相比传统的神经网络，主要的改进是两点：第一是设计了权值共享的卷积层，相比全连接的网络结构，卷积层的局部感知和权值共享减少了参数的数量，降低了网络模型的复杂度。而

第 3 章 基于深度哈希学习的图像检索算法的改进

3.1 引言

目前的深度哈希算法，通过端到端对图像特征提取和二进制码学习，相比传统哈希算法性能有所提升。然而，这些方法仍然存在一些缺点，限制了他们的实际检索性能。首先，相似信息在实际检索系统中通常非常稀疏，即相似图像对的数目远小于不相似图像对的数量。这将导致数据不平衡问题，降低学习到的模型效果。其次，目前大多数深度哈希方法采用sigmoid函数“松弛”的方法来生成连续的近似哈希码，这将导致哈希网络难以收敛，且由于特征经过sigmoid函数之后会损失大量的信息，导致量化误差增大。再次，大多数方法并没有充分利用图像的语义标签信息，图像的语义标签能够提供丰富的语义信息，充分利用语义标签有助于生成语义保留的哈希码，提高系统的检索效率。

本章提出了基于语义保留的深度哈希网络，一种改进的深度哈希学习的网络结构，它主要的创新点是：（1）针对数据不平衡的问题提出了自适应权值的损失函数，减少了数据不平衡带来的影响。（2）针对目前深度哈希学习方法采用sigmoid“松弛”带来的信息损失问题，抛弃了sigmoid“松弛”策略并添加二值约束正则项来减少信息损失和量化误差。（3）为了充分利用图像标签的语义信息，在深度哈希网络后面加入语义保留层，使学习的哈希码保留更多的语义信息。

同时，在语义保留深度哈希的基础上，针对图像数据集有标签数据量少，且获取代价高；而无标签数据量大，易获取却无法被利用的问题。在深度哈希学习中引入生成对抗网络，设计了基于生成对抗网络的半监督哈希学习网络，该网络充分利用无标签数据去提升哈希网络的性能。

3.2 卷积神经网络与哈希学习

卷积神经网络(Convolutional Neural Networks, CNN)是一种典型的神经网络结构，被广泛用于图像识别、目标定位、人脸识别等计算机视觉领域。1998年，Yann LeCun在其论文[37]中首次提出了卷积神经网络，其设计的模型LeNet-5在手写字符识别任务上取得了优异的成绩。CNN相比传统的神经网络，主要的改进是两点：第一是设计了权值共享的卷积层，相比全连接的网络结构，卷积层的局部感知和权值共享减少了参数的数量，降低了网络模型的复杂度。而

且CNN的输入是高维的原始图像像素数据，避免了传统图像分类算法中使用手工提取图像特征。第二点是CNN的池化层设计，使得该网络结构对于平移、旋转、倾斜、比例缩放等具有很强的鲁棒性。同时池化降低了特征图的大小，也减少了参数的数量。这些改进使得CNN能够具有较深的网络层数，但是受制于20世纪初GPU运算单元的发展，早期CNN并没有扩展到很深的网络中。

2012年，随着对CNN研究的深入，和GPU运算单元的发展。Alex Krizhevsky首次提出了经典的CNN模型AlexNet[25]，深度卷积神经网络的意思是指其层数较多。AlexNet提出了很多技术来减少模型参数并加快网络训练，其中relu非线性激活单元的提出解决了之前CNN用sigmoid激活函数一直存在的梯度消失问题。而Dropout的提出解决了在复杂网络模型下CNN容易过拟合的问题。深度卷积神经网络在之后被应用到各种任务上都表现出色。图3-1展示了目前基于深度卷积神经网络的哈希学习的一般网络结构。

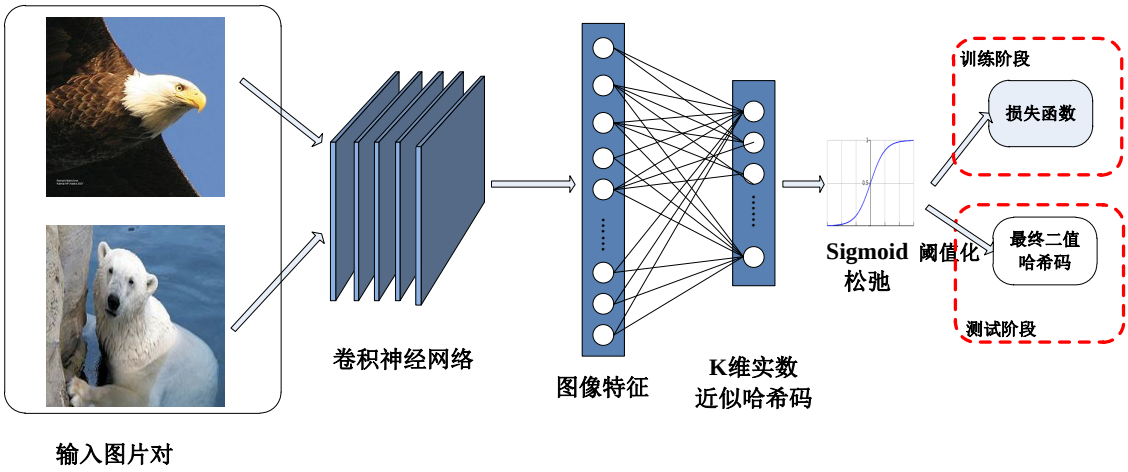

图 3-1 基于卷积神经网络的哈希网络

最左边的是输入图像对，首先是经过卷积神经网络的多次卷积，池化等操作提取图像特征，提取到的高维特征经过后面的全连接层映射为K维实数的近似哈希码。但是哈希的目标是二值的离散哈希码，即每个比特都是$\{0,1\}$，但是如果给这个K维实数的近似哈希码加上离散约束，整个网络就没办法进行反向传播进行更新学习了，所以在之后经sigmoid函数将离散$\{0,1\}$约束松弛，变成$(0,1)$的连续约束，网络就可以进行反向传播和学习了。在测试阶段，哈希码经过sigmoid函数之后再经过阈值化就可以得到最终的二值哈希码。这样端到端的模型把图像特征学习和哈希函数学习融合到一个网络中，提升了整体的性能。

目前基于深度卷积神经网络的哈希学习算法还存在不足，所以为了对其进

行改进，设计了基于语义保留的深度哈希学习模型，实现更高效的深度哈希算法。

3.3 基于语义保留的深度哈希学习

对于基于哈希的检索任务。主要有以下几个方面能够改进检索的效率：

（1）减少数据不平衡的影响。在深度哈希网络训练的过程中，数据集中相似样本数量远远少于不相似样本数量，因为图像对之间的相似必须是属于同一类标签，而数据中类别数往往很多，这就导致数据集中正负样本比例相差很悬殊。比例悬殊的后果是正样本得不到充分训练，而负样本有可能被训练过度导致过拟合影响最后的检索效果，所以要减少数据不平衡对于检索性能的影响。

（2）语义保留。使CNN输出的哈希码尽量保留图像的高层语义信息。也就是语义相似的图片，经过CNN之后输出的哈希码的汉明距离应该比较小。语义不相似的图片，经过CNN之后输出的哈希码的汉明距离应该比较大。如何能在哈希时更好的保留图像的高层语义信息是提升检索性能的关键点。

（3）减少量化误差。在图像检索的过程中引入哈希主要的原因是计算哈希码的汉明距离比计算高维实数特征的欧氏距离要快的多。要快速的计算汉明距离，哈希码必须要求是二值的。但是CNN的反向传播优化算法只能处理连续的数据，不能处理二值的数据。所以在使用CNN进行哈希学习时，一般的做法是通过sigmoid函数将实数压缩到0到1之间，再经过量化得到最终的哈希码。所以减少实数的近似哈希码到二值哈希码之间的量化误差，能使信息损失的最少，得到的哈希码检索性能也更好。

基于语义保留的深度哈希学习模型的整个网络结构如图3-2所示，该网络由四个关键部分组成：（1）标准的卷积神经网络（CNN），例如AlexNet和ResNet，用于学习深层图像表示。（2）用于将图像特征变换为低维二值哈希码的全连接层。（3）用于解决数据不平衡问题的自适应权值的损失函数，和减少量化误差的二值约束正则项，也称作量化损失。（4）哈希层之后的语义保留层，学习语义保留信息，提升性能。

3.3.1 网络结构

输入的图像对首先进入CNN中提取图像特征，CNN中包含一组堆叠的卷积层。数据在经过每一个卷积层时，会有多个不同的卷积核对其进行卷积，不同

的卷积核对不同的特征敏感。每经过一个卷积层能得到图像的一组特征。经过多个卷积层之后，得到的特征逐渐从低层次的特征（颜色，形状等）变成了高层次的语义特征。

表 3-1 CNN网络具体结构

层类型	卷积核大小	步长	输出尺寸	输出个数
输入层	*	*	32x32	BS
卷积层	5x5	2	32x32	32
池化层	3x3	2	15x15	32
卷积层	5x5	2	15x15	32
池化层	3x3	2	7x7	32
卷积层	5x5	2	7x7	64
池化层	3x3	1	3x3	64
全连接层	*	*	32x32	500
全连接层	*	*	32x32	K

提取到图像特征之后，再经过一个全连接层，将特征映射为哈希码。但是由于CNN不能进行离散优化，于是得到的其实是近似的K维实数值哈希码，需要进行量化操作才能得到最终的二值哈希码。

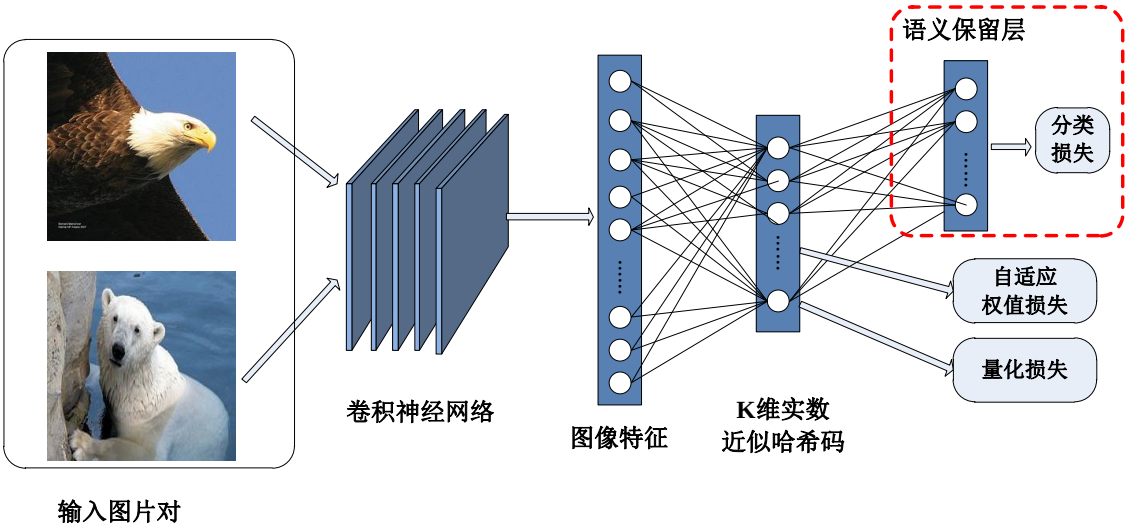

图 3-2 语义保留的深度哈希学习网络结构

本章模型中的CNN结构是来自论文DSH[38]，网络模型的参数如表3-1所示。表3-1中第一行的BS，代表BatchSize，即一次训练迭代输入图像的数量。符

号*表示该层没有这个参数。最后一行的K表示K比特哈希码，这个数字是可变的，常用的有16，32，48，64等。

图3-2最右上部分是语义保留层，K维连续的哈希码经过语义保留层的线性映射，再进行分类损失的计算，反向更新整个网络的权值。同时K维连续的哈希码也直接计算自适应权值损失和量化损失，同样的进行反向传播更新网络。

3.3.2 自适应权值损失函数

在获得图像的近似哈希码之后，需要进行计算损失函数，进行反向传播更新网络权值。损失函数计算的是当前哈希码与学习目标的差距，损失函数设计的好坏直接决定了最后学习到的哈希码的好坏，所以损失函数的设计非常重要。在目前的深度哈希学习算法中，很多算法的损失函数都没有考虑到数据不平衡的问题，数据不平衡是指输入图片中相似图片对远小于不相似图片对的数量。例如，数据输入的一个batch大小是200，即200张图片，10类，每类有20张；两张图片之间类标相同则相似，否则不相似。所以这200张图片两两组成图片对，可以组成的相似图片对数量是：10*（20*19/2）=1900，而不相似的图片对数量是：200*199/2-1900=18000。正负样本的比例达到1: 9.47，正样本偏少会导致训练过程过度关注负样本，导致正样本学习不充分，负样本学习容易过拟合，降低整体的检索性能。为了解决这个问题，本节提出了一个自适应权值的损失函数，根据正负样本比例自动调节正负样本损失值权值，使的整个网络的学习更充分。下面是具体的设计思想。

给定训练师数据集X，包含N个数据点$\{x_i\}_{i=1}^{N}$，哈希学习的目标是学习一个从X到k比特二值码的映射：

$$X \to \{-1,+1\}^k \tag{3-1}$$

该映射使得相似的图片被编码成相似的二值码。相似图片的二值码汉明距离应该尽量接近，而不相似图片的二值码汉明距离应该尽量远。基于这个目标，设计的损失函数应该拉近相似图片的二值码汉明距离，推远不相似图片的二值码汉明距离。设S代表相似矩阵，$s_{ij}=1$代表数据x_i和x_j相似，$s_{ij}=0$代表数据x_i和x_j不相似。图片对的相似是根据它们的类标信息来确定的，当它们拥有相同的类标则相似，否则不相似。需要特别指出的是，对于多标签图像数据，两张图片至少有一个相同的类标就视为相似。

对于一个图像对$x_i, x_j \in X$和其对应的网络二值输出$b_i, b_j \in \{-1,+1\}^k$，对应这个图像对的损失函数为：

$$\begin{aligned} L(b_i, b_j, s_{ij}) = & \frac{1}{2} s_{ij} D_h(b_i, b_j) \\ & + \frac{1}{2}(1 - s_{ij}) \max(m - D_h(b_i, b_j), 0) \\ & s.t. \quad b_i, b_j \in \{+1, -1\}^k, i, j \in \{1, 2..., N\} \end{aligned} \tag{3-2}$$

其中$D_h(\cdot\,,\cdot)$代表两个二值向量的汉明距离，$m > 0$是一个阈值参数。公式的第一项惩罚相似的图片对被映射到不相似的二值码，第二项惩罚不相似的图片被映射到相似的二值码（相似的定义是汉明距离小于阈值m）。

由于存在数据不平衡，所以可以为相似样本的惩罚力度和不相似样本的惩罚力度设置不同的值，即加上一个权值w_{ij}，w_{ij}能根据不同数据量大小变化。

$$w_{ij} = \begin{cases} 1/|S_1|, s_{ij} = 1 \\ 1/|S_0|, s_{ij} = 0 \end{cases} \tag{3-3}$$

其中$S_1 = \{s_{ij} \in S : s_{ij} = 1\}$是一次训练batch中所有的相似图片对，$S_0 = \{s_{ij} \in S : s_{ij} = 0\}$是所有的不相似图片对。加上权值之后的损失函数为:

$$\begin{aligned} L(b_i, b_j, s_{ij}) = & w_{ij}(\frac{1}{2} s_{ij} D_h(b_i, b_j) \\ & + \frac{1}{2}(1 - s_{ij}) \max(m - D_h(b_i, b_j), 0)) \\ & s.t. \quad b_i, b_j \in \{+1, -1\}^k, i, j \in \{1, 2..., N\} \end{aligned} \tag{3-4}$$

然而，对于这样一个损失函数，由于$b_{i,j} \in \{+1, -1\}^k$，是离散的，无法进行神经网络的反向传播，所以，我们将公式（3-4）中汉明距离替换成了欧式距离，并去掉了离散约束。公式（3-4）被改写成

$$\begin{aligned} L(b_i - b_j, s_{ij}) = & w_{ij}(\frac{1}{2} s_{ij} \left\| b_i - b_j \right\|_2^2 \\ & + \frac{1}{2}(1 - s_{ij}) \max(m - \left\| b_i - b_j \right\|_2^2, 0)) \end{aligned} \tag{3-5}$$

从公式（3-3）我们可以看出，由于相似样本数量少，则$|S_1| < |S_0|$。当样本$s_{ij} = 1$，即为相似样本对时，对应的权值$w_{ij} = 1/|S_1|$，要小于$s_{ij} = 0$时即不相似样本对所对应的权值$w_{ij} = |S_0|$。在加上自适应的权值w_{ij}之后，相似样本对的损失值被放大，而不相似样本对的损失值相应缩小，平衡了因为样本数目差距带来的影响。

3.3.3 二值约束正则项

目前深度哈希算法都存在一个sigmoid松弛层，它的目的是将实数域压缩到0到1之间，本质上是为了减少之后量化的误差，但是sigmoid函数本身会导致网络训练时梯度消失，导致网络训练不充分；而且由于将输出范围压缩

在0到1之间的连续值，会损失大量的图像信息，导致最后得到的哈希码对于图像检索效果不够好。所以本文算法去掉了sigmoid松弛层，在损失函数中加入一个二值约束正则项去减少量化误差，保证近似哈希码值靠近-1或者+1。这个正则项的表达式为$f(x) = ||x| - 1|$，该函数如图3-3所示。

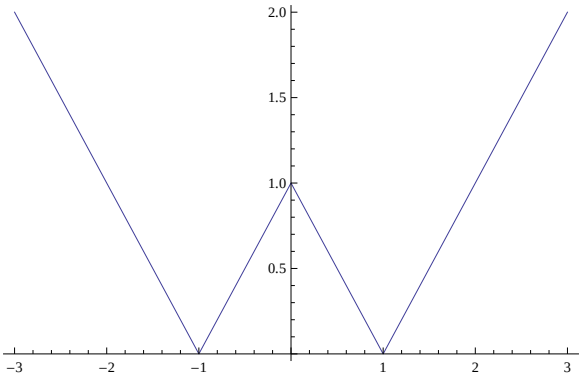

图 3-3 正则项函数图像

由函数图像可以看出，要使正则项损失尽量小的话，哈希码应该尽量接近-1或是+1，在网络不断训练过程中，哈希码会不断向-1和+1靠近。通过正则项的方式减少量化误差，不会产生梯度消失的问题，能使网络训练的更充分，学习到更好的哈希码。将原始损失函数加上正则项之后的损失函数为：

$$\begin{aligned} L_r(b_i, b_j - s_{ij}) = w_{ij}(&\frac{1}{2} s_{ij} \left\| b_i - b_j \right\|_2^2 \\ &+ \frac{1}{2}(1 - s_{ij}) \max(m - \left\| b_i - b_j \right\|_2^2, 0)) \\ &+ \alpha(\| \, |b_i| - 1 \, \|_1 + \| \, |b_j| - 1 \, \|_1) \end{aligned} \tag{3-6}$$

其中下角标r代表加了正则项的损失函数，α是超参数，控制正则项的约束力度。$\|\cdot\|_1$是对于向量的L1正则，$|\cdot|$ 是对向量逐元素的取绝对值操作。而对于所有的图片对而言，我们的最终目标是最小化这个全局损失函数：

$$\ell_r = \sum_{s_{ij} \in S} L_r(b_i, b_j, s_{ij}) \tag{3-7}$$

值得一提的是，由于正则项函数在（-1，0）和（+1，0）点不可导，所以在反向传播时采用次导数的算法，这两个点的次导数区间是[1, 1]，次导数可以为该区间的任意数，于是我们定义该点的次导数为1，这样就可以使用标准的梯度下降算法来更新网络权值了。公式3-6对应$b_{i,j}, \forall i, j$的梯度计算如下所示：

$$\begin{aligned}
\frac{\partial Trem1}{\partial b_{i,j}} &= w_{ij}\cdot(-1)^{j+1}(1-s_{ij})(b_i-b_j)\\
\frac{\partial Trem2}{\partial b_{i,j}} &= w_{ij}\cdot\begin{Bmatrix}(-1)^j s_{ij}(b_i-b_j), & \left\|b_i-b_j\right\|_2^2 < m\\ 0, & \left\|b_i-b_j\right\|_2^2 \geqslant m\end{Bmatrix}\\
\frac{\partial Trem3}{\partial b_{i,j}} &= \alpha\varphi(b_{i,j}),\ \varphi(x)=\begin{Bmatrix}1, & -1\leqslant x\leqslant 0\ or\ x\geqslant 1\\ -1, & otherwise\end{Bmatrix}
\end{aligned} \tag{3-8}$$

有了这些计算的梯度值，就能够进行梯度下降反向传播优化网络权值。在获得近似哈希码b_i之后，再通过阈值化函数$sign(b_i)$就可以得到最终需要的二值哈希码。这种正则项约束的方法不使用*sigmoid*函数去近似量化过程，不会有梯度消失的问题，可以加快网络的训练速度。

3.3.4 语义保留层

图像类别标签不仅能提供分类信息，也是一种对哈希学习很有用的监督信息。本节对标签和哈希码之间的关系进行建模，以构建语义保留的二值哈希码。我们先假设学习到的哈希码很好的保留了语义信息，那么保留有语义信息的哈希码对于分类任务来说也应该是很好的特征。这意味着我们可以用图像的哈希码来对图像进行分类，通过优化分类损失，确保语义相似的图片能映射到相似的哈希码。

如图3-2中所示，我们在近似哈希码之后加了一层语义保留层（红色虚线框）。输出节点数是m，它代表数据集的类别数，让权值矩阵$W\in R^{k\times m}$表示将哈希码映射为类标的线性映射。对每张图片$x_i,\ i\in 1,2,...,N$，其对应的近似哈希码为b_i，分类预测为$\hat{y}_i$，而图片的真实类标为y_i，y_i采用one-hot的表达形式，举例来说，如果数据集有10类，有一张图片属于第2类，则其对应的one-hot标签为：$[0,1,0,0,0,0,0,0,0,0]$，即除了第二个位置为1其他位置都为0，这张图片属于哪个类哪个位置就为1，这样所有的图片真实类标都可以表示为一个等长的向量，即使这张图片对应多个类别。为了学习权值矩阵W并更新前面的哈希网络，只需要不断优化以下的目标函数：

$$\ell_s = \arg\min_{W}\sum_{i=1}^{N} L(y_i,\hat{y}_i)+\lambda\left\|W\right\|_2^2 \tag{3-9}$$

这个目标函数是要使得用哈希码进行分类预测得到的类标值要和真实的类标值相同，其中$L(\cdot)$代表预测值和真实值的误差度量，λ是控制正则项重要程度的参数。

（1）单标签分类

对于单标签数据集的多分类，误差度量函数可以选择softmax多分类损失函数，具体形式是：

$$L(\mathrm{y}_i, \hat{y}_i) = -\sum_{j=1}^{m} y_{ij} \ln \hat{y}_{ij} \tag{3-10}$$

其中y_{ij}是第i张图片的第j个输出单元的期望输出（即第i张图片属不属于第j个类），因为真实类标是one-hot编码，所以$y_{ij} \in \{0, 1\}$。$\hat{y}_{ij}$是第i张图片的第j个输出单元的预测输出（即第i张图片属于第j个类的概率）。$\hat{y}_{ij}$形式如下：

$$\hat{y}_{ij} = \frac{e^{w_j^T \cdot b_i}}{\sum_{k=1}^{m} e^{w_k^T \cdot b_i}} \tag{3-11}$$

有了$\hat{y}_{ij}$的形式就可以计算$L(y_i, \hat{y}_i)$对于b_i的梯度值了，具体的计算公式如3-12。

$$\frac{\partial L(y_i, \hat{y}_i)}{\partial b_i} = -\sum_{j=1}^{m} y_{ij}(w_j - \frac{\sum_{k=1}^{m} w_k e^{w_k^T b_i}}{\sum_{k=1}^{m} e^{w_k^T b_i}}) \tag{3-12}$$

（2）多标签分类。

softmax函数对于单标签分类有很好的效果。但要对多标签进行分类的话就不能使用softmax，因为softmax只适用于单标签的分类。对于多标签多分类任务，目前比较常用的是多分类的逻辑回归，即对每个类别和其他类别构造一个二分类，对m分类就会有m个交叉熵损失。于是对于多标签数据集多分类进行误差度量函数形式为：

$$L(\mathrm{y}_i, \hat{y}_i) = -\sum_{j=1}^{m} (y_{ij} \ln \hat{y}_{ij} + (1 - y_{ij}) \ln(1 - \hat{y}_{ij})) \tag{3-13}$$

其中

$$\hat{y}_i = \begin{bmatrix} \sigma(w_1^T \cdot b_i) \\ \sigma(w_2^T \cdot b_i) \\ \cdots \\ \sigma(w_2^T \cdot b_i) \end{bmatrix}, \ \sigma(x) = \frac{1}{1 + e^{-x}} \tag{3-14}$$

根据以上的公式可以计算多标签分类时$L(y_i, \hat{y}_i)$对于b_i的梯度值:

$$\frac{\partial L(y_i, \hat{y}_i)}{\partial b_i} = -\sum_{j=1}^{m} y_{ij}(1 - \sigma(w_j^T b_i))w_j + (1 - y_{ij})\sigma(w_j^T b_i)w_j \tag{3-15}$$

根据不同的数据集使用不同的分类损失函数计算梯度。在网络加入语义保留层之后，整个网络的全局损失函数为：

$$\ell_{all} = \ell_r + \ell_s \tag{3-16}$$

有了最终的全局损失函数，就可以用批量梯度下降进行反向传播优化网络参数。

3.4 基于生成对抗网络的半监督哈希学习

深度神经网络一般都需要在大量有标签的数据集上进行训练，才能取得比较好的效果。但是现实情况是由于人工标注数据需要消耗大量的财力物力，导致有标签的数据量相对无标签的数据量要少很多，大量的无标签数据无法被利用。于是半监督学习的概念被提出来，半监督学习能够同时利用有标签数据和无标签数据提升模型的性能，对于有标签数据很少而无标签数据很多的情况非常适用。而最近研究火热的生成对抗网络模型，被证明对于半监督学习任务有很好的效果[39, 40]。同样的，对于深度哈希学习任务，也存在大量没有标签的数据，借鉴GAN用于半监督学习的思想，本文提出了基于生成对抗网络的半监督哈希学习算法，充分利用无标签数据，提升检索的性能。

3.4.1 生成对抗网络

2014年Goodfellow等人提出的一种新型的神经网络模型，生成对抗网络（GAN，Generative adversarial networks）[41]，它是一种概率生成模型。GAN的设计思想来自于博弈论中的零和博弈，即博弈双方的收益和为零，一方的收益必然会导致另一方的损失。GAN的结构可以分为两部分，生成器和判别器，相当于博弈的两方。生成器的工作是负责学习真实样本的数据分布，并生成符合分布的新数据；而判别器的工作是负责鉴别输入的数据是真实数据还是由生成器生成的数据。GAN让生成器和判别器之间不断地竞争，从而分别提高两个模型的生成能力和判别能力。它的结构如图3-4所示。

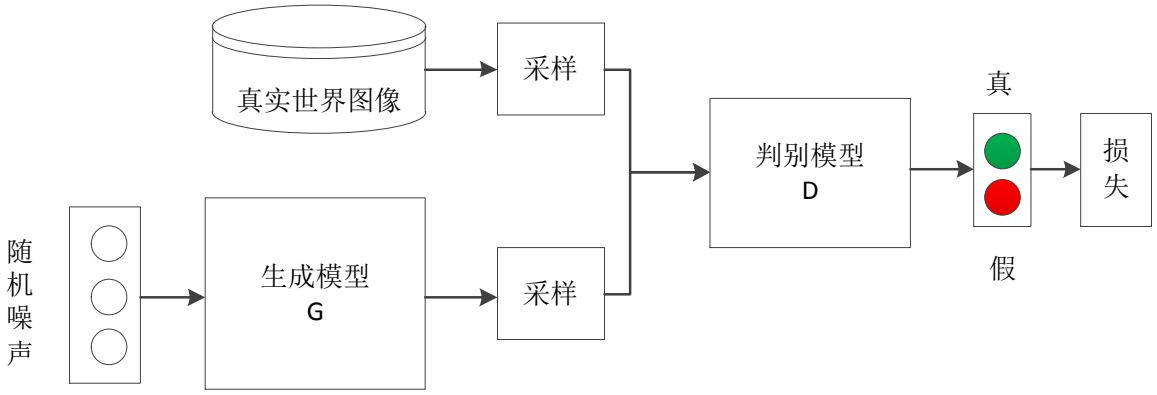

图 3-4 GAN的结构

GAN的生成器和判别器都能用深度神经网络实现。生成器的输入是服从均匀分布的随机噪声，输出是和真实图像一样大小的图像。判别器接受两个输入，一个是真实图像，一个是生成器生成的图像。GAN训练时，先固定生成器

的权值训练判别器参数，再固定判别器权值训练生成器参数，交替迭代，直到达到一种动态的平衡，即纳什均衡[42]。在训练的过程中，双方都极力优化自己的网络。下面分别是生成器和判别器的损失函数:

$$\begin{aligned} J^{(D)} &= -\frac{1}{2}\mathbb{E}_{x\sim p_{data}} \log D(x) - \frac{1}{2}\mathbb{E}_z \log(1 - D(G(z))) \\ J^{(G)} &= -J^{(D)} \end{aligned} \tag{3-17}$$

其中D代表判别器，G代表生成器，z是随机噪声。它们的损失函数正好相反。最后GAN能获得一个性能出色的判别器，同时获得一个能生成真实样本分布数据的生成器。

GAN是一个无监督模型，它只需要真实图像数据，不需要任何人工的标注数据，所以它的适用范围很广，因为互联网上存在大量的未标注数据，GAN可以很好的利用这些数据。GAN目前已经成为深度学习研究领域的一个重要研究方向。目前，GAN已经应用到了各个领域，在图像生成领域GAN已经可以生成人脸，数字，和各种各样的物体和场景图片[43]。同时在超像素重建[44]、图像风格转换[45]、自然语言处理[46]等领域GAN都有出色的表现。

3.4.2 设计思想

由于GAN不需要监督数据，适用于半监督哈希学习。半监督学习（Semi-Supervised Learning, SSL）是介于无监督学习和有监督学习之间的一种学习算法，它同时利用了少量的有标签数据和大量无标签数据。在半监督学习的框架中，需要N个独立同分布的样例$x_1, x_2, ..., x_N \in X$和对应的标签$y_1, y_2, ..., y_N \in Y$。另外，同时还需要M个无标签的样例$x_{N+1}, x_{N+2}, ..., x_{N+M} \in X$。半监督学习利用这种信息的组合来超越可以通过丢弃未标记数据进行有监督学习或丢弃标签数据进行无监督学习而获得的学习性能。

半监督学习的作用如图3-5所示，展示的半监督学习用于分类任务上的效果。上面的方框中有两个数据点，一个数据点是正例（白色空心圆），另一个是负例（黑色实心圆），中间的虚线表示监督学习得到的分类界面。很明显由于数据量过少，学习到的分界面不够好，泛化能力弱。而下图的方框中是采用半监督方法得到分类界面结果，其中灰色点是大量的无标签样本，在同时考虑有标签数据和无标签数据之后，模型能更好的理解数据的整体分布，得到分类效果更好的分界面。这可以被视为执行聚类，用标签数据标记聚类，将决策边界推离高密度区域，使得模型的泛化能力更强。

利用GAN进行半监督学习的原理如图3-6所示。因为无标签样本不能够直接进行训练，于是引入GAN，GAN的生成器可以从随机信号中生成“虚假”样

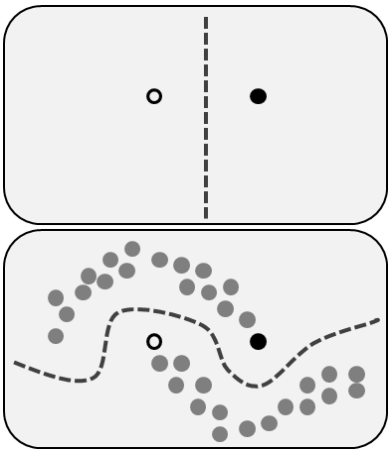

图 3-5 半监督学习

本，而数据中的有标签和样本和无标签样本都可以认为是“真实”样本，这样就相当于给原本没有标签的无标签样本赋予了一个“真”的标签（左边的大椭圆），给从生成器生成的数据赋予了一个“假”的标签（右边的小椭圆），从而可以对无标签样本进行学习。由于拥有这样的“真假”标签，GAN还有一个判别器对训练样本进行区分真假，额外的，对于有标签的样本，同时也进行有监督的训练。

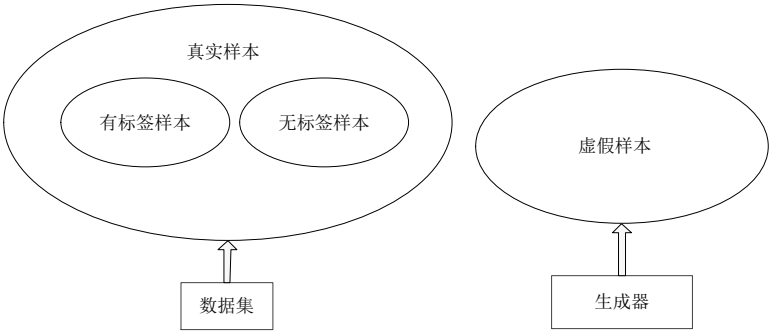

图 3-6 GAN用于半监督学习

半监督学习相比监督学习效果更好的原因是半监督学习能够利用更多的数据学习，从中学习到的数据的分布有助于生成更具有泛化能力的模型。举个通俗的例子，对于学习汉字这个任务，有监督学习就是同时给你汉字和对应的读音和含义，而半监督学习就是在有监督学习的基础上，同时还多给你很多的汉字，但是并没有告诉对应的含义和读音。多出来的汉字就算没人教，多练习区分“是否是汉字”，也对学习汉字有帮助。因为这在某些程度上给予了一定的汉字相关分布信息。

3.4.3 网络结构

鉴于GAN在半监督学习上的优秀表现，为了充分利用无标签数据，提升深度哈希网络的性能，本节结合GAN的思想，在深度哈希网络中加入生成器进行半监督学习，将深度哈希网络当做GAN的判别器，同时训练生成器和判别器，提升最终整体的性能。

图3-7是基于生成对抗网络的半监督哈希学习网络的具体结构，在之前的结构上新增加了一个生成器，生成器的输入是随机生成的均匀分布噪声，经过多层堆叠的反卷积层生成与真实图像相同大小的图片。这里生成器的结构参照DCGAN[47]结构。网络中的激活函数都为relu，且对每一层生成的特征图再进行Batch Normalization[48]操作，使模型不易崩溃，并加快网络收敛速度。图3-7中的判别器和之前的哈希网络略有不同，它加入了一个判别器节点，该判别器节点专门为判别图片是否为真图片设定，同时对应的有判别损失，来指导生成器和判别器的优化。

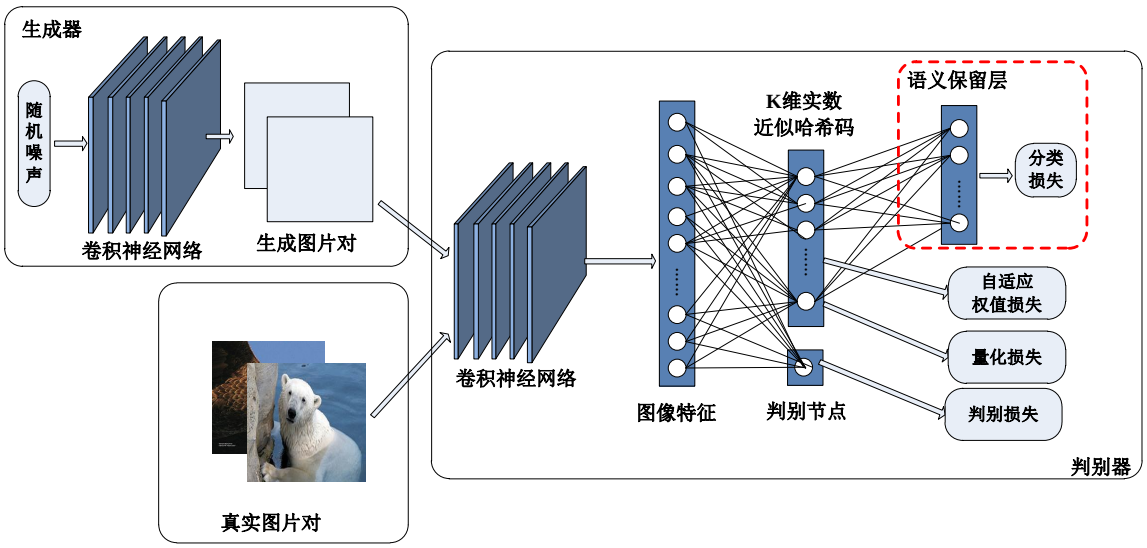

图 3-7 基于生成对抗网络的半监督哈希学习网络

训练分为两个阶段，判别器更新阶段和生成器更新阶段。首先当需要更新判别器时，需要固定生成器权值，生成器生成“假”的图片对，有了“真”图片对和“假”图片对，就可以计算哈希学习损失，“真”图片判别的损失，“假”图片判别的损失，三者加起来共同优化判别器网络。而当生成器需要更新时，固定判别器权值，使“假”图片判别的损失上升并更新生成器权值。不断的迭代训练，直到最后达到稳定状态。

在测试阶段，则不需要生成器的参与。只需要将真实图片经过判别器，得

到对应K维近似哈希码，再经过阈值化得到最后的哈希码。

3.4.4 生成器结构

在GAN中，生成器设计的目标是从随机噪声中生成满足真实样本分布的虚假样本，使得判别器难以分清真实样本和虚假样本。最早GAN提出时，生成器的结构是由普通的全连接神经网络组成，但是这样的模型训练容易崩溃，本文采用的是卷积神经网络来实现的生成器，具体结构如表3-2所示。

表 3-2 生成器具体结构

层类型	卷积核大小	步长	输出尺寸	输出个数
输入层	*	*	100	BS
全连接层	*	*	2x2	512
反卷积层	5x5	2	4x4	256
反卷积层	5x5	2	8x8	128
反卷积层	5x5	2	16x16	64
反卷积层	5x5	2	32x32	3

表中符号*表示该层没有这个参数，BS表示batchsize大小。反卷积层其实也是一种卷积层，只不过是将卷积的前向传播和反向传播逆了过来[49]。特征图经过反卷积层之后会变大，所以初始的100维均匀分布噪声经过生成器之后扩展为了32*32*3的大小，和真实的图片大小相同。

对于生成器来说，它的优化目标是尽量让判别器无法分辨出真实图片和生成器生成的图片。所以它的目标函数要尽量让判别器误判的概率高。即它的目标可以分为：（1）使得真实样本被判别为负类；（2）使得生成的样本被判别为正类。在判别器中存在一个专门的判别节点，所以生成器的目标函数为：

$$L_G = \mathbb{E}_{x\sim p_{data(x)}} logD(x) + \mathbb{E}_{z\sim noise} log(1 - D(G(z))) \tag{3-18}$$

其中$x \sim p_{data(x)}$表示真实样本分布，$z \sim noise$表示虚假样本分布，$G(z)$表示由随机噪声经过生成器生成的图片，$D(x)$表示图片经过判别器被判别为真图片的概率。生成器通过不断优化这个目标函数，使得判别器无法辨别出真假图片。

3.4.5 判别器结构

由图3-7可以看出，基于生成对抗网络的半监督哈希学习网络中判别器是由两部分构成的，一部分是前面所提出的基于语义保留的哈希网络，另一部分

是判别节点，目的是判别真实和虚假图片。具体的，判别器接收两个输入，一个是随机噪声经过生成器生成的图片，另一个输入是真实数据集中的图片。在输入经过卷积神经网络提取特征之后，再经过全连接层，得到了$K+1$个节点的输出，其中前K个节点，是哈希网络输出的K维近似哈希码，而第$K+1$个节点是判别节点。K维近似哈希码训练时会计算语义保留层的分类损失、自适应权值损失和量化损失。而判别节点会计算对应的判别损失。

GAN用于半监督哈希学习，判别器的目标是有两个，一个是优化有标签数据的哈希损失，另一个是优化判别节点使得判别器更加的精准。对于第一个目标，是同时需要真实数据和对应的标签信息的，而对于第二个目标，只需要无标签的数据即可，所以可以按这个将这两个目标将目标函数分为有监督的和无监督的，具体的联合目标函数为：

$$L_D = L_{supervised} + L_{unsupervised} \tag{3-19}$$

其中，

$$\begin{aligned} L_{supervised} &= \ell_{all} \\ L_{unsupervised} &= -\{\mathbb{E}_{x\sim p_{data(x)}} logD(x) + \mathbb{E}_{z\sim noise} log(1-D(G(z)))\} \end{aligned} \tag{3-20}$$

ℓ_{all}是前面计算的基于语义保留的哈希损失，具体参见公式3-16。$L_{unsupervised}$正好和L_G相反，体现了对抗生成网络的无监督思想，同时$L_{supervised}$体现了监督哈希学习的思想，两者结合训练就是一种半监督哈希的思想。无标签数据在训练时不计算$L_{supervised}$。

3.5 本章小结

本章针对目前深度哈希学习中存在的问题，提出基于语义保留的深度哈希学习算法。主要的改进为：（1）设计了自适应权值的损失函数解决数据不平衡问题。（2）去掉了sigmoid松弛层，并采用了二值约束正则项减少哈希学习的量化误差。（3）添加语义保留层，使哈希码保留语义信息，提升检索性能。

同时本章针对大规模数据集中有标签数据集量少，不易获取，而无标签数据量大，易获取却无法被充分利用的问题，提出了基于生成对抗网络的半监督哈希学习算法。该算法通过在输入层加入一个生成器，生成“虚假”图片，从而使得有标签样本和无标签样本都具有“真实”图片的标签，并通过在网络输出加入判别节点对其进行区分，使得无标签样本能够被充分学习，提升哈希网络的性能。

第 4 章 实验设计与分析

4.1 引言

本章的主要内容是通过实验评估基于语义保留的深度哈希算法和基于生成对抗网络的半监督哈希算法的性能。通过对比其他的深度哈希算法和传统的哈希算法，验证本文提出算法的有效性。

4.2 实验数据

本章实验使用两个图像数据集进行测试，分别是CIFAR-10[50]和NUS-WIDE[51]数据集。

（1）CIFAR-10

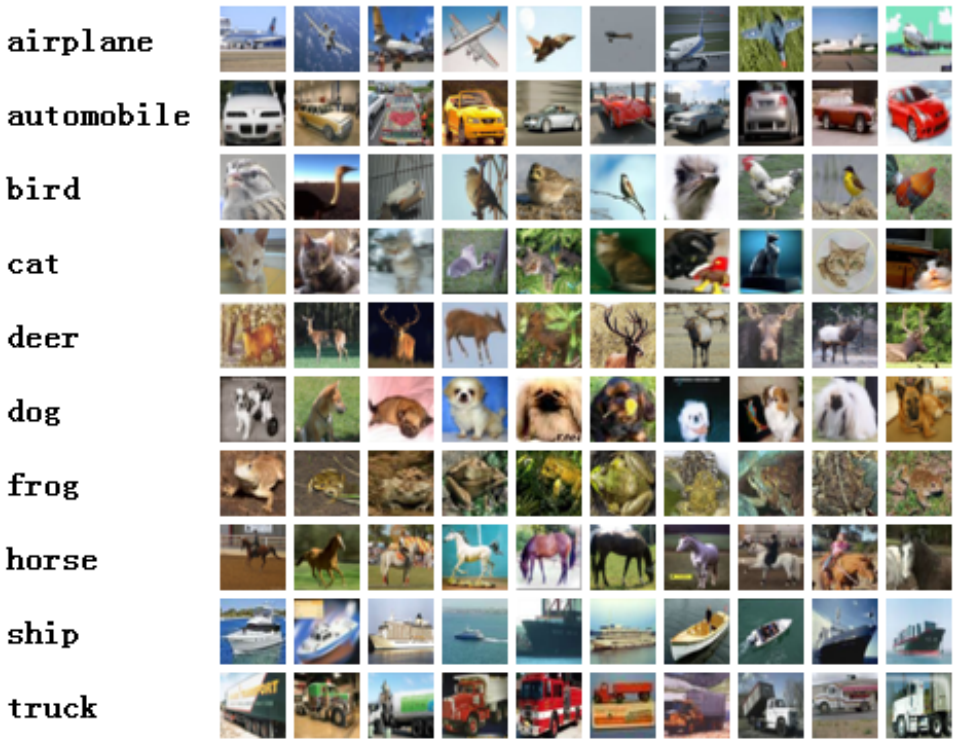

图 4-1 CIFAR10数据集

CIFAR-10是AlexNet的提出者Alex Krizhevsky收集的一个有标注的彩色图像数据集。CIFAR-10数据集包含60000张32x32的RGB彩色图片，该数据集总共