基于局部特征的图像表示模型理论与实践

（申请清华大学工学博士学位论文）

培 养 单 位 ： 计 算 机 科 学 与 技 术 系

学　　科 ： 计 算 机 科 学 与 技 术

研 究 生 ： 谢 凌 曦

指 导 教 师 ： 张　　钹 教 授

二〇一五年六月

Image Representation Models based on Local Features: Theory and Practise

Dissertation Submitted to

Tsinghua University

in partial fulfillment of the requirement

for the degree of

Doctor of Philosophy

in

Computer Science and Technology

by

XIE Lingxi

Dissertation Supervisor : Professor ZHANG Bo

June, 2015

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：

清华大学拥有在著作权法规定范围内学位论文的使用权，其中包括：（1）已获学位的研究生必须按学校规定提交学位论文，学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文；（2）为教学和科研目的，学校可以将公开的学位论文作为资料在图书馆、资料室等场所供校内师生阅读，或在校园网上供校内师生浏览部分内容；（3）根据《中华人民共和国学位条例暂行实施办法》，向国家图书馆报送可以公开的学位论文。

本人保证遵守上述规定。

（保密的论文在解密后应遵守此规定）

作者签名：_____ 导师签名：_____

日　　期：_____ 日　　期：_____

摘 要

图像的分类和检索一直是计算机视觉、模式识别和机器学习领域的核心课题之一。基于局部特征的图像表示模型是图像分类和检索中最为有效和常用的方法。然而，由于众所周知的语义鸿沟，以及局部特征在表达高级视觉概念方面的局限性，传统的图像表示模型通常存在许多缺陷，包括对细微噪声的敏感性太强、特征编码的空间结构信息缺失、无法捕捉实际问题中的兴趣区域，等等。

本文针对这些问题进行了充分的研究，讨论和改进了两种基于局部特征的图像表示模型，即视觉词袋模型和深度卷积神经网络。从实际应用的角度出发，我们抽象出若干重要的科学问题，并且利用创新性的方法解决这些问题。我们将图像表示模型拆分为多个模块，包括特征抽取、特征编码、特征组合以及后处理等，逐一进行深入探索。在模块化研究的基础上，我们开创性地提出一种能够同时处理图像分类和检索任务的模型，完成了两者的统一。最后，我们提出了两个富有挑战性的计算机视觉新问题，并且提供了新颖的初步解决方案。

本文的主要创新点包括以下六个方面：

- 提出一种局部特征强化算法：从图像分类和检索的实际情况出发，论述局部特征的翻转不变的必要性，并且设计了一种简单的解决方案。
- 提出一种利用空间位置信息强化特征编码的算法：通过构造几何视觉短语和基于短语的池化算法，使得特征编码具有描述局部特征组的能力。
- 提出两种图像空间匹配模型：针对特定图像分类问题（细粒度分类和场景分类）的特殊特征组合算法，提升了图像表示的质量。
- 提出两种针对图像检索问题的后处理算法：利用基于图结构和随机游走理论的扩散算法，大幅提升准确率和召回率，并且应用于大规模网络图像搜索。
- 提出一种统一的图像分类和检索模型：利用强有力的图像表示和鲁棒的距离计算方法，同时处理分类和检索问题，并且在两类任务上都达到先进水平。
- 提出两个计算机视觉领域的新问题：同时利用前面几章的研究成果以及创新性的框架结构，对新问题进行探索，并且提出了初步的解决方案。

本文所提出的方法大多具有很强的推广性，能够很方便地移植到其他应用问题中。我们的研究，为计算机视觉领域的科研人员提供了许多有价值的线索；我们提出的有趣而富有挑战性的新问题，也为我们未来的研究工作奠定了基础。

关键词：计算机视觉，局部特征，图像表示，图像分类，图像检索

Abstract

Image classification and retrieval have been core problems in computer vision, pattern recognition and machine learning. Image representation based on local features is the most popular approach in image classification and retrieval. However, due to the well-known semantic gap and the limitations of local features in representing high-level visual concepts, conventional image representation models often suffer from a lot of shortcomings, including the sensitiveness to small noises, the lack of spatial structure information in feature encoding, the difficulty in capturing regions-of-interest in specified classification and/or retrieval problems, *etc*.

This thesis presents extensive research efforts to combat the above issues, providing insightful improvements and discussions to two types of image representation models based on local features, namely the Bag-of-Visual-Words (BoVW) model and the deep Convolutional Neural Network (CNN). Starting from real-world applications, we abstract several important scientific problems and suggest novel solutions. We partition image representation into several modules and explore each one of them in depth, including feature extraction, feature encoding, feature summarization, post-processing, *etc*. Based on modular research, we propose a pioneering unified model to deal with both image classification and retrieval problems. Finally, we suggest two challenging problems in computer vision, and provide primary yet innovative approaches to deal with them.

The main innovations of this thesis are summarized in the following six aspects.

- We propose an algorithm for local feature enhancement. Based on the observation in real-world classification and retrieval problems, we demonstrate the importance of reversal invariance of local features, and then design a straightforward solution.
- We propose an algorithm which applies spatial information to enhance feature encoding. With the construction of "geometric visual phrases", we embed more powerful descriptive power into the encoded features.
- We propose two spatial matching algorithms to cope with two specified classification problems, *i.e.*, fine-grained object recognition and scene recognition, and improve image representation quality.
- We propose two post-processing algorithms for image retrieval and large-scale Web image search. Based on the graph-based data mining algorithms and the random

walk theory, both the precision and recall of image retrieval are significantly improved.

- We propose a unified model for both image classification and retrieval, which is on the basis of powerful regional features and robust computation of the image-to-class distance. To our knowledge, it is a very first trial towards unifying these two problems, which also achieves state-of-the-art performance.
- We propose two challenging research topics in computer vision, and provide elementary efforts based on state-of-the-art techniques and innovative organization structures. These works might pave a new way to future researches in the computer vision community.

One major contribution of this thesis lies in the powerful generalization ability of the proposed methods. Most of them could be applied to various problems, *i.e.*, not limited to the evaluated cases, and produce consistent improvements. Our research provides several new clues for researchers on the related research fields. The proposed interesting yet challenging problems also lay the foundation of our future works.

Key words: Computer Vision; Local Features; Image Representation; Image Classification; Image Retrieval

目 录

主要符号对照表

CV	计算机视觉（Computer Vision）
ML	机器学习（Machine Learning）
PR	模式识别（Pattern Recognition）
BoVW	视觉词袋（Bag-of-Visual-Words）
CNN	卷积神经网络（Convolutional Neural Networks）
SIFT	尺度不变特征变换（Scale Invariant Feature Transform）
LCS	局部颜色统计量（Local Color Statistics）
SPM	空间金字塔匹配（Spatial Pyramid Matching）
SVM	支持向量机（Support Vector Machine）
DPM	可变形的部位模型（Deformable Part Model）
ANN	近似近邻（Approximate Nearest Neighbor）
RIDE	翻转不变特征强化（Reversal Invariant Descriptor Enhancement）
GPP	几何短语池化（Geometric Phrase Pooling）
GRSP	推广的规则空间池化（Generalized Regular Spatial Pooling）
HPM	层次化部件匹配（Hierarchical Part Matching）
OPM	朝向金字塔匹配（Orientational Pyramid Matching）
HGP	异质图传播（Heterogeneous Graph Propagation）
IQE	增量查询扩展（Incremental Query Expansion）
IFV	图像特征投票（Image-Feature Voting）
ImageWeb	图像网络
ONE	在线最近邻估计（Online Nearest-neighbor Estimation）
$\mathcal{I}$	图像数据集
N	图像数据集样本数
C	图像数据集的类别数
$\mathbf{I}_n$	第n个图像数据
W, H	图像的宽和高（像素数）
$\mathcal{P}$	图像像素集
$\mathcal{D}$	描述子集合

M	描述子个数
$\mathbf{d}_m$	第m个描述子的描述向量
D	描述向量的维度
$\mathbf{l}_m$	第m个描述子的位置向量
$\mathcal{R}_m$	第m个描述子的几何区域（像素集）
$\mathcal{B}$	视觉码本
B	视觉码本的大小
$\mathcal{W}$	特征向量集合
$\mathbf{w}_m$	第m个特征向量
$\mathcal{J}$	描述子集合（特征集合）的索引集
S	索引集的子集个数
$\mathcal{J}_s$	索引集的第s个子集
$\mathbf{f}_s$	第s个区域的图像表示向量
$\mathbf{F}$	图像的表示向量

第 1 章 引言

本文研究的主要内容是基于图像表示的图像分类和检索任务。这是计算机视觉领域最基本的问题之一。本章首先介绍研究工作的背景和现实意义，随后阐明这一领域的现状和主要困难，最后描述文章的整体结构和创新点。

1.1 研究背景和现实意义

在信息量迅速增长的今天，人类的活动正在源源不断地产生大量的信息，特别是多媒体信息。 2012年，Google搜索引擎宣布已经索引超过30万亿张网页①。假设平均每张网页上有1张图片，即使按照正常人每秒钟能够浏览10张图片来估计，一个人浏览完这些图片也需要超过3000年的时间。 2014年，Youtube网站宣布平均每分钟用户上传视频超过300个小时②，这意味着每天上传到Youtube的视频需要一个人花费50年的时间才能看完——这个数字比2013年时增加了3倍，并且还将继续增长。诚然，海量的图片和视频并非对每个用户都具有价值。因此，如何从中筛选出有意义的内容推荐给用户，成为了搜索引擎和视频网站的迫切需求。另一方面，在这些图片和视频中，难免存在一些反动、色情、暴力或者恐怖主义发布的内容。在以往，这些信息可以通过人工检查来筛除；但是随着信息量的高速增长，政府和公司迫切地需要一种快速的自动算法，代替人力完成繁重的筛查任务。 2014年，中央网络安全和信息化领导小组举办了第一届特定音视频分析系统评测资格大赛③，以期利用图像处理技术来辅助审查和筛除互联网上流传的恐怖主义视频。

作为人工智能的一个重要分支，计算机视觉主要研究如何帮助计算机系统获取并理解图像和视频信息，进而拥有与人类视觉相当的信息处理能力。图像的分类和检索是计算机视觉领域最核心的课题之一：图像分类问题要求系统从有限的标注图像中学习视觉概念，并且用于判断未标注图像的类别信息；图像检索问题则要求系统在短时间内寻找查询图像的近似样本，包括局部近似和全局近似。这两个问题不仅代表着计算机视觉的研究前沿，也蕴涵着大量的实际应用，受到了学术界和工业界的广泛关注。例如，基于ImageNet数据集的大规模视觉识别竞

① http://en.wikipedia.org/wiki/Google_Search
② http://www.youtube.com/yt/press/statistics.html
③ http://avid.erangelab.com

赛① 受到广泛的关注，同时以Google图像搜索② 为代表的大规模商业图像搜索引擎也在不断发展。

1.2 研究现状和难点

上个世纪四十年代，克劳德·香农③ 提出现代信息理论（Morden Information Theory）[1]，奠定了数据在计算机内部的存储和传输基础。这一理论被沿用至今，决定了图像在计算机内的存储方式（即0/1数据串）不具有语义特征的基本事实，从而导致了图像的高级语义概念无法直接被计算机所理解。

为了更加有效地获取图像中的语义信息，传统的计算机视觉算法通常将图像表示为一个长向量的形式，并且在此基础上进行分析。遵循这种思路，视觉词袋模型被广泛应用于图像分类[2]和检索[3]问题。典型的视觉词袋模型具有一定的相似性，即利用局部特征，自底向上地构建复杂的概念和语义。由于原始像素信息通常无法包含足够的语义信息，人工设计的局部特征（Local Features）[4][5]被提取出来，并且用于构建视觉字典（Visual Vocabulary）[6][7]，以将局部特征编码为具有固定长度的视觉单词（Visual Words）[8][9]。随后，这些视觉单词被组合起来，成为图像级的特征向量（Feature Vectors）（主要用于图像分类任务[10]），或者形成倒排表（Inverted Index，主要用于图像检索任务[3]），以便后续查询操作。另外近年来，基于深度学习（Deep Learning）理论的卷积神经网络（Convolutional Neural Network，CNN）也被广泛应用于图像分析，如大规模图像分类[11]，物体检测[12]及迁移特征提取[13]。从本质上看，卷积神经网络[14]可以视为传统的感知器模型的一个变种。它允许通过简单的函数（如线性卷积函数，池化函数和线性激励函数等）来构建复杂的神经网络模型，用于拟合大样本的特征空间分布。

然而，传统的图像表示模型具有一些无法回避的缺陷，如熟知的语义鸿沟（Semantic Gap）[15][16]。在许多先前的工作中，研究者已经发现：这些缺陷通常由局部特征的不稳定性，如同义性和多义性造成[17]。为了跨越语义鸿沟，研究者们提出了各种解决方案，包括提取多种局部特征[18][19]、构建视觉短语[17][20][21]、构建层次化的空间池化算法（针对分类问题）[22][10]、后处理和重排序（针对检索问题）[6][23][24]，等等。然而，如何将这些模块组合成为一个完整的系统使得它们能够互相配合，仍然是一个开放性的问题。

① http://www.image-net.org/challenges/LSVRC/

② http://images.google.com

③ 克劳德·香农（Claude Shannon），1916–2001，美国数学家，信息论创始人。他在论文中提出，“将布尔代数应用于电子领域，能够构建并解决任何逻辑和数值关系”，从而奠定了现代信息理论的基础。

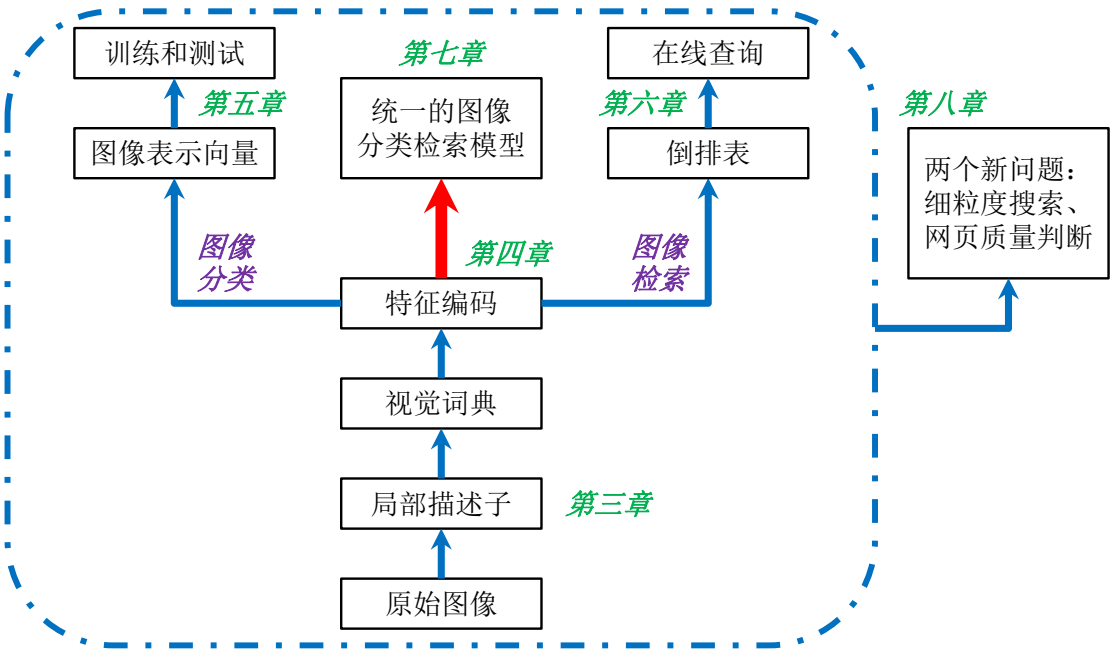


图 1.1　视觉词袋模型的总体结构以及本文的主体框架。

1.3　全文的结构和创新点

本文将基于局部特征的图像表示模型分成若干模块，分别对其中存在的缺陷提出针对性的解决方案。我们将探索一个高效的图像分类/检索模型应当具备的特性，如局部特征的多样性和稳定性，视觉概念描述的层次性以及空间组织结构的精确性。基于这些观察，我们提出了一系列算法，以提高图像分类和检索的精度。我们的研究表明：为了实现精确的分类和检索，不仅需要通用性的算法（如第4章提出的几何视觉短语池化，用于增强局部特征编码的表达能力），也需要针对特定任务设计的算法（如第5章提出的，用于细粒度分类和场景分类的空间切分方法）；不仅需要在局部特征尺度进行改进（如第3章提出的，具有翻转不变性的局部描述子），也需要充分挖掘全局特征的描述能力（如第7章提出的，统一的分类和检索算法）。通过这些研究工作，我们获得了许多适合图像分析的线索。这些线索不仅使我们的算法达到了当前领先的准确率，也能够对未来的研究工作产生一定的启发。图1.1展示了全文的总体结构。

1.3.1　本文的组织结构

全文分为九章，组织如下：

- 第1章为引言，概括本文的研究内容和主要贡献。
- 第2章为背景知识，介绍本文所需要的基础知识，包括视觉词袋模型和卷积神经网络。

- 第3三章介绍一种局部特征强化的方法，为局部特征提供翻转不变的特性。我们将从一个简单的例子出发，逐步将我们的方法推广到一般的情形，并且应用于更多的实际问题中。
- 第4章介绍一种利用空间位置信息强化特征编码的算法，为局部的特征编码提供更强大的描述能力。通过提出视觉短语的概念以及针对视觉短语的特征组合算法，我们大幅提升了传统的稀疏编码模型的表示能力。
- 第5章介绍两种针对特定分类问题的特征组合方式。在细粒度分类问题和场景分类问题上，我们分别采取基于物体部件的空间匹配和基于表面朝向的空间匹配算法，以提升图像表示的质量。
- 第6章介绍两种针对检索问题的后处理算法，以大幅提升检索的准确率和召回率。我们将检索问题规划为图的信息检索，并从经典的随机游走模型出发，寻求应用于大规模图像搜索的快速而有效的算法。
- 第7章介绍一种基于全局图像表示的模型，能够统一地处理图像分类和检索问题。我们充分利用深度卷积神经网络模型的表示能力和图形处理单元的并行运算能力，使得算法的分类和检索性能均达到当前先进水平。
- 第8章提出几个计算机视觉研究中的新问题。我们将利用现有方法以及一些创新性的组织结构，对这些问题进行初步的探索。
- 第9章对本文的内容进行总结，并且对未来的工作做出展望。

1.3.2 本文的主要创新点

本文的主要贡献在于从多种视角来观察和研究基于局部特征的图像表示模型。通过在不同模块上提出的改进方案，我们能够在所研究的问题上取得良好的效果，同时也容易将提出的方法移植到实际应用中去。我们的研究为相关领域的科研人员提供了许多线索，同时也为未来的工作奠定了基础。

本文的主要创新点包括以下六个方面：

- 提出一种局部特征强化算法：从图像分类和检索的实际情况出发，论述局部特征的翻转不变的必要性，并且设计了一种简单的解决方案。
- 提出一种利用空间位置信息强化特征编码的算法：通过构造几何视觉短语和基于短语的池化算法，使得特征编码具有描述局部特征组的能力。
- 提出两种图像空间匹配模型：针对特定图像分类问题（细粒度分类和场景分类）的特殊特征组合算法，提升了图像表示的质量。
- 提出两种针对图像检索问题的后处理算法：利用基于图结构和随机游走理论的扩散算法，大幅提升图像检索的准确率和召回率，并且应用于大规模网络图像搜索。

- 提出一种统一的图像分类和检索模型：利用强有力的图像表示和鲁棒的距离计算方法，同时处理分类和检索问题，并且在两类任务上都达到先进水平。
- 提出两个计算机视觉领域的新问题：同时利用前面几章的研究成果以及创新性的框架结构，对新问题进行探索，并且提出初步的解决方案。

第 2 章　背景知识

在开始正式的讨论之前，我们先介绍一些背景知识。这将有助于所研究问题的引入和展开。

2.1　人工智能与计算机视觉

人工智能（Artificial Intelligence，AI）的研究目标是使人工制造的系统（通常指计算机系统）表现出一定的自主行为能力。它涉及到许多不同的研究领域，如计算机视觉（Computer Vision，CV）、自然语言处理（Natural Language Processing，NLP）、决策理论（Decision Theory）、专家系统（Expert Systems）、遗传算法（Genetic Algorithms）、机器人学（Robotics）等。在计算机系统中实现的人工智能，往往依赖于一定程度的先验知识，以及从大量的训练样本中学习获得的某种模式（Pattern）。作为一个应用广泛且具有挑战性的学科分支，计算机视觉吸引了大量科研人员和商业公司的兴趣，成为人工智能领域最重要、最热门的研究课题之一。

众所周知，人类从外界接收的信息中，有超过80%是通过视觉获得的。然而，计算机的数据存储方式决定了电子图片中无法包含语义信息，因此让计算机学会如何“看”是一个具有挑战性的课题。

作为一门“新兴”学科，计算机视觉的起源可以追溯到1966年，马尔文·闵斯基[①]将计算机视觉作为本科生的一门暑期课程。到上世纪70年代，计算机视觉已经能够对一些特定的图像进行简单的处理（如边缘检测）。上世纪90年代开始，计算机视觉在某些特定的问题（如人脸识别、行人识别、车牌检测等）上取得了较高的准确率，逐渐开始成为一门具有实用价值的工程学科。进入21世纪，随着计算机运算能力的飞跃以及新模型（如视觉词袋模型、卷积神经网络等）的提出，计算机视觉能够解决的问题不断增加，鲁棒性不断增强，在许多任务（如人脸识别）上甚至超越了人类的识别精度。同时，随着图像处理技术的不断成熟，许多算法也被移植到视频处理应用中，形成了良好的发展态势。

2.1.1　典型问题

计算机视觉的典型问题包括：图像分类、图像检索、物体检测、图像分割、

① 马尔文·闵斯基（Marvin Minsky），1927–，美国认知学家，1969年图灵奖（Turing Award）获得者。著有《感知器》（Perceptrons）一书。

边缘检测、三维视觉，等等。下面，我们针对每个问题进行简要的介绍。

2.1.1.1 图像分类

图像分类（Image Classification）的目标是将一些图片按照所包含的视觉概念进行分类。早期的图像分类任务中通常只包含有少数几个类别，如确认图像是否包含相应概念的二分类问题[25]、为手写输入设备服务的光学字符识别（Optical Character Recognition，OCR）[26]、以及针对少量视觉概念以及限定条件下图片的小规模识别问题[27]。近年来，随着图像分类技术的不断发展和成熟，分类任务的类别数量出现了明显的增长。成百[28][29]上千乃至上万[30]类的视觉概念已经能够被有效地识别。同时，图像分类任务还从一般视觉概念的区分程度，细化为针对某些特定视觉概念的细粒度区分，例如场景分类[10][31]、特定植物分类[32][33]、特定动物分类[34][35]和人造物品分类[36]等。

2.1.1.2 图像检索

图像检索（Image Retrieval）通常需要索引一个较大的图像数据库，并且根据用户的查询请求返回相关联的图片。本文所研究的图像检索，一般指基于内容的图像检索（Content Based Image Retrieval，CBIR），即不根据图片的元信息（Meta Information，如周围文本等），只根据图片本身的内容进行检索。早期的图像检索算法受到硬件设备的限制，通常只能索引较小的数据库[37]，并且完成一些较为简单的查询请求（如寻找颜色或者形状较为接近的图片），而且对噪声的抵抗性较差。从图像相关性的定义上看，图像检索任务可以大致分为两类，即近似重复图像检索（Near-duplicate Image Retrieval）和部分重复图像检索（Partial-duplicate Image Retrieval）。近年来，局部特征的逐渐成熟使得部分重复图像检索的重心从全局特征[37] 转向局部特征[6][20][38]，倒排表[3]的应用也使得检索时间相比于数据库大小呈次线性增长[39]。同时，全局特征的优化和改进也为大规模近似重复图像检索提供了提升尺度的空间[40]。早在2010年，Google图片搜索引擎就已经检索了超过100亿张图片①。如今，一些基于商品信息匹配的图像检索技术也已经上线，如CamFind② 和Snap Fashion③。

2.1.1.3 物体检测

物体检测（Object Detection）的目标是找出图像中包含的物体，并且将它们

① http://en.wikipedia.org/wiki/Google_Images
② http://camfindapp.com
③ http://www.snapfashion.co.uk

的位置标示出来。物体检测与图像分类问题具有很强的联系，两者可以互相配合以提高识别/检测的效果：对图像内容进行识别有助于提高检测的精度，而事先检测出兴趣物体的位置也有利于提高识别的准确率[41]。早期的物体检测任务通常针对某些特定的物体[5]，如人脸、车辆、行人，等等。随着检测技术的进步，能够有效检测的物体种类不断增加。基于一般物体识别算法的模型，甚至可以检测出一般物体[12]。物体检测问题的主要难度在于目标可能具有不同的姿态（pose），并且在局部出现变形（deformation）和遮挡（occlusion）等噪声。同时，如果被检测物体的尺度较小，也会对算法造成一定的困难[42]。

与物体检测相关的另外一个任务是*物体再确认*（Object Re-identification）。宽泛地说，“再确认”任务通常标定一个物体（如监控视频里的某个人物），要求系统从其他图像库（视频帧）中找到同样的物体。如果将每个物体看做一个*实例*（instance），那么再确认的任务设定与图像检索就存在某些相似之处[43]。物体再确认技术可以应用于许多实际问题，如监控视频里的人物和车辆的跟踪[44]。

2.1.1.4 图像分割

图像分割（Image Segmentation）的目标是根据语义信息，将图像切割为若干部分。最初的图像分割方法应用于前背景切分问题[45][46]。对于每张图像，需要提供像素级别的输出信息，即明确指定每一个像素属于前景或者背景。对于分割结果的评价可以通过简单的逐像素分类准确率计算，但是由于背景所占区域通常较大，所以这种算法对于背景有所偏置。为了防止这种偏置，可以通过改进的评估算法，计算*接收者操作特征曲线*（Receiver Operating Characteristic curve，ROC curve）的下面积[47]来评价分割结果。

在前背景分割的基础上更进一步，可以将图像分成若干不同的区域，即*图像语法分析*（Image Parsing）。例如对于场景图像，可以找出天空、地面、建筑物、行人等不同的成分[48][49]。近年来，还出现了不少商业应用，研究如何对人的衣着进行切分，即*衣着语法分析*（Clothing Parsing）[50][51]。在Fashionista数据集[50]中，甚至出现了56个不同的类别标签，如头发、面部、上下衣、鞋袜、手提包甚至戒指等细微物品。

2.1.1.5 边缘检测

边缘检测（Edge Detection或Boundary Detection）是计算机视觉领域最早研究的问题之一。边缘检测的目的是标识图像中的某些位置，这些位置的两侧具有语义上较强的变化。边缘通常反映了某些重要的事件或者变化，包括深度上的不连续、表面方向不连续、物质属性变化、场景照明变化，等等。

边缘检测并不容易。按照一般的判断标准，可以将边缘定义为图像亮度变化比较强的位置，因此边缘检测就与计算亮度的梯度相关。然而，除非场景中的照明得到很好的控制并且物体表面的反射特性非常稳定，否则很容易造成误判。边缘检测中值得注意的一类问题是边界检测（boundary detection）[52]：它不仅要求找出明显的边缘，还需要这些边缘形成闭合图形，从而包含更强的语义信息。

边缘检测可以应用于许多其他的问题。例如，可以利用闭合的边缘结构（也称为边界）进行图像分割[53]，将物体切分为多个部件；此外，边缘检测还有助于分析图像的形状特征[18]，从而对物体识别提供额外的信息。

2.1.1.6 三维视觉

三维视觉（Three-Dimensional Vision，3D视觉）泛指将传统计算机视觉在二维图像上的任务扩展到三维图像上（立体图形）。三维视觉更加接近于人类观察物体的方式，也被认为是计算机视觉的未来发展方向。由于三维图像的处理方式通常有别于二维图像，许多新的技术也由此得到了发展，例如用于三维图像描述的三维描述子[54]以及三维物体分类[55]等。

2.1.2 相关研究领域

计算机视觉与许多其他研究领域紧密相关，包括机器学习、模式识别、机器感知、信号处理等。

- 机器学习（Machine Learning，ML）的主要研究内容是设计一些让计算机主动学习的算法。通常，机器学习算法从数据中自动分析获得规律，并利用规律对未知数据进行预测。机器学习的许多算法都能够有效地辅助计算机视觉应用，包括支持向量机（Support Vector Machine，SVM）[56]、多层感知器（Multi-Layer Perceptron，MLP）[57][58]、概率图模型（Probability Graph Model）[59]，等等。
- 模式识别（Pattern Recognition，PR）使用数学的技术和方法来研究模式的自动处理和判读。计算机视觉的基本任务，就是从视觉输入中发现一定的模式，再将这些模式用于新数据的分析。由于机器学习与模式识别的任务有很大部分的重叠[60]，在不引起混淆的情况下，两者可以混合使用。
- 机器感知（Machine Perception，MP）主要指计算机通过分析数据以达到与人类相似的感受能力。人类的感受能力主要有视觉、听觉、嗅觉、味觉、触觉等，计算机视觉研究计算机的视觉感受能力。
- 信号处理（Signal Processing，SP）泛指对信号的表示、变换、运算等进行处理的过程。从广义角度看，计算机视觉也被称为视觉信号处理。虽然

视觉信号具有某些特殊性，但是经典信号处理的算法也经常被用于计算机视觉领域，如滤波（filtering）、傅立叶分析（Fourier analysis）、降采样（sub-sampling），等等。

计算机视觉的发展能够促进这些研究领域的前进，而这些领域的许多算法也被应用于计算机视觉以改进其效果。

2.2 视觉词袋模型

视觉词袋模型（Bag-of-Visual-Words model，BoVW model），也称特征包模型（Bag-of-Features model，BoF model），是一种利用局部特征的提取、编码和组合来表示图像的方法[3][2]。它也是得到最广泛应用的传统图像表示模型之一。

视觉词袋模型的主要流程是：描述子抽取、视觉码本训练、特征编码、特征组合（用于图像分类）和特征索引（用于图像检索）。下面分别介绍这些模块。

2.2.1 描述子抽取

视觉词袋模型的输入是一张图像（image），表示为$\mathbf{I} = \left(a_{ij}\right)_{W\times H}$。其中$W$和$H$是图像的宽度和高度，$a_{ij}$表示灰度图像的灰度值，或者彩色图像的色彩空间向量。

由于像素点对于图像的表达能力有限，手工设计的描述子（descriptors）通常被用于描述局部区块（patch）特征。描述子是一种算子（operator），它们作用的图像区域称为兴趣区域（Regions of Interest，RoI）。对于兴趣区域的寻找，通常采用基于梯度信息的局部最大值检测算法。典型的算法包括高斯差分（Differential of Gaussian，DoG）[4]、*Hessian*或*Harris*仿射变换（Hessian/Harris Affine，HA）[61]、最稳定外部区域（Maximally Stable Extremal Region，MSER）[62]、密集特征点（Dense Interest Points，DIP）[63]、密集采样（dense sampling）[64]，等等。特别地，对于图像分类问题，密集特征点和密集采样算法往往能够达到更好的效果。对于局部特征的描述，常见的例子包括尺度不变特征变换（Scale Invariant Feature Transform，SIFT）[4]、有向梯度直方图（Histogram of Oriented Gradients，HOG）[5]、梯度的位置和朝向直方图（Gradient Location and Orientation Histogram，GLOH）[65]、加速的鲁棒特征（Speeded Up Robust Features，SURF）[66]、二值鲁棒的独立基本特征（Binary Robust Independent Elementary Features，BRIEF）[67]、*DAISY*描述子（DAISY descriptors）[68]、有向*FAST*和旋转*BRIEF*特征（Oriented FAST and Rotated BRIEF，ORB）[69]，等等。这些特征具有不同的适用范围，被广泛应用于分类、检索、检测等任务中。

除了上述纹理特征外，其他特征（如颜色和形状）也经常被提取出来以加强图像的表示。一种常见的提取颜色特征的方法，是在多个不同的*颜色信道*（color channels）上提取纹理特征。根据*色彩空间*（color space）的不同，提取的特征也不同，如RGB-SIFT、HSV-SIFT、RGB-HOG等。基于基本的色彩空间（如RGB），也可以人工设计一些信道分量，并且在这些分量上提取特征。典型的例子如C-SIFT、Opponent-SIFT等[70]，它们都被证明比简单的RGB-SIFT更加有效。其他专门针对颜色设计的特征，如*局部颜色统计量*（Local Color Statistics，LCS）[7]，也能够有效地描述颜色信息，从而辅助纹理特征对图像进行表示。类似地，对形状特征的描述，可以通过在边缘图像上提取纹理特征，如*边缘SIFT*（Edge-SIFT）[71]，或者设计专门的形状描述子，如*形状上下文*（Shape Context，SC）[72]，*内部距离形状上下文*（Inner Distance Shape Context，IDSC）[73]。多种特征通过不同的方式融合，以实现对图像不同方面进行描述的目的[18]。

无论通过何种检测和描述方法，最终都能够得到一个局部特征的集合：

$$\mathcal{D} = \{(\mathbf{d}_1, \mathbf{l}_1), (\mathbf{d}_2, \mathbf{l}_2), \ldots, (\mathbf{d}_M, \mathbf{l}_M)\} \tag{2-1}$$

其中，$\mathbf{d}_m$和$\mathbf{l}_m$是第m个描述子的描述向量（D维向量）和平面位置，M是描述子的总数。如果需要用到描述子的更加丰富的几何信息（如尺度和形状），可以利用像素集合$\mathcal{R}_m$代替位置向量$\mathbf{l}_m$。在图像分类和检索任务中，每张图像通常会提取成百甚至上千的描述子。如果利用多种描述子，则上述集合可能不止一个[71]。

2.2.2 视觉码本训练

在局部描述子被提取出来后，需要有一种方法来捕捉它们在特征空间中的分布情况。此时，通常训练一个*视觉码本*（visual codebook），或*视觉词典*（visual vocabulary），以利用*核密度模型*（kernel density function）来近似表达空间的分布情况。视觉码本是一个由多个*视觉单词*（visual words，一些与描述子具有相同维度的向量）构成的集合：

$$\mathcal{B} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_B\} \tag{2-2}$$

其中，$\mathbf{c}_b$是第b个视觉单词。 B是视觉单词的数目，也称为视觉码本的大小。在特征空间中，每个描述子都与其最近邻的若干视觉单词有关。上述码本通常利用*K均值聚类*（K-Means clustering）来计算。聚类计算需要反复进行最近邻查询操作，精确的聚类算法的时间开销比较大。因此在聚类规模较大时，往往采用*层次化K均值聚类*（Hierarchical K-Means，HKM）[74] 或者*近似K均值聚类*

（Approximate K-Means，AKM）[6]以降低时间开销。相关研究也表明，在小规模码本上采用嵌入（embedding）方法也能够达到类似大码本的效果[75]。

除了K均值聚类，其他算法如高斯混合模型（Gaussian Mixture Model，GMM）也可以用于描述特征空间的分布情况。由于增加了方差信息，GMM通常能够描述更加丰富的空间分布情况。以K个高斯分布混合而成的模型为例：

$$\mathcal{B} = \{(\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \ldots, (\pi_B, \boldsymbol{\mu}_B, \boldsymbol{\Sigma}_B)\} \tag{2-3}$$

其中，π_b、$\boldsymbol{\mu}_b$和$\boldsymbol{\Sigma}_b$分别表示第b个分布的先验概率、均值和方差。GMM可以利用期望最大化（Expectation Maximization，EM）[76]算法迭代获得。在实际应用中，通常将$\boldsymbol{\Sigma}_b$限制为对角矩阵以降低训练复杂度[7]。

2.2.3 特征编码

特征编码（Feature Encoding）过程能够将特征编码为固定长度的紧凑（compact）向量，以便后续处理。

如果视觉码本由K均值算法聚类获得（由多个视觉单词组成），那么每个描述子便可以由其相邻的一个或多个视觉单词编码。硬量化（Hard Quantization，HQ）利用距离一个描述子最近的视觉单词来编码它。在图像检索问题中，硬量化也被称为向量量化（Vector Quantization，VQ）。这种量化方式虽然简单，但一般存在较大的量化误差。作为另一种量化方式，软量化（Soft Quantization，SQ）允许利用多个视觉单词来编码一个描述子。稀疏编码（Sparse Coding，SC）[77]就是其中的一个例子，它的一些实例如稀疏编码-空间金字塔匹配（Sparse coding Spatial Pyramid Matching，ScSPM）[78]和局部限制的线性编码（Locality-constrained Linear Coding，LLC）[9]都是有效的图像编码算法。在编码后，描述子$\mathbf{d}_m$被表示为一个B维向量$\mathbf{w}_m$，其每一维分量对应于每个视觉单词的响应。通常，$\mathbf{w}_m$是一个稀疏向量，只有极少数几个分量为非零值。

如果视觉码本由GMM表达，保存了较详细的几何信息，那么可以利用*Fisher*向量（Fisher Vectors，FV）来表达更加丰富的图像内容[79]。Fisher向量通过对*Fisher*信息矩阵（Fisher Information Matrix，FIM）的近似分解获得[80]。通过对GMM的先验（0阶信息）、均值（1阶信息）和方差（2阶信息）进行编码，可以记录更多的图像统计量。从而得到一个更长（$(2D+1)B$维）、更密集（超过50%非零分量）的图像表示向量。类似的思路也被用于其他高维特征编码，如超向量编码（Super Vector encoding，SV encoding）[81]和有向特征编码（Oriented SIFT/HOG encoding）[82]等。

在图像检索任务中，还可以采用一类*无码本量化*（codebook-free quantization）方法。典型的例子如*标量量化*（Scalar Quantization，SQ，对应于向量量化）[83]。对于一个D维描述子（如SIFT），$\mathbf{d}_m = (d_{m,1}, d_{m,2}, \ldots, d_{m,D})$，可以直接定义一个阈值（如描述子分量的中位数），并且相应地二值化描述子，得到一个0/1*位向量*（bit vector）。利用更加精细的阈值划分，也可以将向量量化（二值化）为更长的位向量。以此法量化后，特征之间的相似性就可以通过位向量之间的*海明距离*（Hamming distance）来衡量。

特征编码完毕后，局部描述子的集合便转化为一个特征集合：

$$\mathcal{W} = \{(\mathbf{w}_1, \mathbf{l}_1), (\mathbf{w}_2, \mathbf{l}_2), \ldots, (\mathbf{w}_M, \mathbf{l}_M)\} \tag{2-4}$$

这里，$\mathbf{w}_m$代替了(2-1)式中的$\mathbf{d}_m$。术语方面，$\mathbf{d}_m$被称为第m个描述向量，而$\mathbf{w}_m$则是对应于$\mathbf{d}_m$的特征向量。然而在不引起歧义的情况下，我们将不加区分地使用“特征”来指代“描述子”，如“SIFT特征”。

在特征编码过程后，视觉词袋模型的工作流程根据具体任务而有所不同。

2.2.4 特征组合和图像分类

在分类任务中，往往需要将一张图像表示为一个长向量。将(2-4)式中的M个特征综合为一个长向量的过程称为*池化*（pooling）。池化过程对于视觉词袋模型非常重要：因为它具有抵消平移变换的效果，从而能够识别出现于图像不同位置的关键特征。

一种自然的池化方法是计算M个特征的某种全局统计量。*最大池化*（max-pooling）和*平均池化*（average-pooling）或许是应用最为广泛的两种算法。最大池化计算每一维（对应于视觉单词）的最大响应：$\mathbf{f} = \max_{1\le m\le M} \mathbf{w}_m$；而平均池化则计算每一维的平均响应：$\mathbf{f} = \frac{1}{M}\sum_{m=1}^{M} \mathbf{w}_m$。这里，记号$\max_m$和$\sum_m$表示逐维的求最大值和求和运算。最大池化和平均池化的区别已经被充分讨论[84]。理论上，最大池化能够更好地与软量化算法配合，而平均池化则更适合硬量化算法。*推广的最大池化*（Generalized Max Pooling，GMP）[85]则讨论了最大池化与Fisher向量的关系。最大池化和平均池化都是p-*范数池化*（ℓ_p-norm pooling）的特例。p-范数池化的一般表达式为：$\mathbf{f} = \left[\sum_{1\le m\le M} \mathbf{w}_m^p\right]^{1/p}$，其中$\mathbf{w}_m^p$表示逐维对向量$\mathbf{w}_m$进行$p$次幂运算。当$p \to +\infty$和$p = 1$时，p-范数池化分别退化为最大池化和平均池化。不同的p的取值能够产生不同的池化效果。除了手工调节以外，还可以利用某种优化函数，对每张图像寻找最优的p值，如*几何p-范数池化*（Geometric ℓ_p-norm Pooling，GLP）算法[86]。

全局的池化算法忽略了图像中丰富的空间位置信息，而这些信息可能对图像的理解非常有用。基于核匹配理论（kernel matching theory），研究者提出了能够编码空间位置信息的空间池化（spatial pooling）算法，如金字塔匹配（Pyramid Matching，PM）[22] 和空间金字塔匹配（Spatial Pyramid Matching，SPM）[10]。它们通过将图像切分为若干小区域，来建模图像的上下文内容。显式地，令$\mathcal{J} = \{1, 2, \ldots, M\}$表示集合$\mathcal{W}$中特征的索引（index）集合。空间池化算法定义$S$个$\mathcal{J}$的子集，标记为$\{\mathcal{J}_1, \mathcal{J}_2, \ldots, \mathcal{J}_S\}$，并且对于每个子集分别进行池化，得到$S$个池化向量：$\{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_S\}$。 SPM算法对于$\mathcal{J}$的切分依赖于一种层次化的网格结构。除了这种简单的方法，还可以通过更加灵活的空间切分来达到更好的效果[87][88]。对于细粒度分类问题，选择能够描述物体部件（part）信息的池化集合至关重要。许多算法[89][90][91]通过检测、分割等方式来达到这一目的。

池化算法后，可以得到S个独立的向量：$\{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_S\}$。这些向量可以具有相同[10]或者不同[92]的维度。视觉词袋模型的最后一个模块，是将这些向量进行归一化（normalization）以消除不同特征之间数值范围的影响。最常见的归一化方式是p-范数归一化（ℓ_p-norm normalization），定义为将向量投影到特征空间中的p-范数球面上：$\widetilde{\mathbf{F}} = \mathbf{F} / \|\mathbf{F}\|_p$。 p的选择对分类结果影响很大[9]。对于支持向量机，2-范数归一化被证明具有最小的结构风险（structural risk）[93]。之后，归一化的向量被连接起来作为视觉词袋模型的最终输出：$\mathbf{F} = [\mathbf{f}_1; \mathbf{f}_2; \ldots; \mathbf{f}_S]$。归一化和连接操作的先后顺序也能显著地影响分类结果：如果切分后的S个子集具有独立性，那么应当先分别对S个向量进行归一化，再进行拼接[7]。此外，考虑每个子集的重要性并且对其进行加权，也能够提升图像表示的效果[94]。

视觉词袋模型的输出是一个长向量，其维度通常能够达到10^5甚至10^6级别[19]。由于训练样本的个数相对较少，能够有效避免过拟合的支持向量机（Support Vector Machine，SVM）成为最常用的分类器。近年来，随着图像数据集不断增大[30]，可扩展性（scalability）在实际应用中越来越重要。一般来说，训练一对一（one-vs-one，OVO）分类器比训练一对它（one-vs-rest，OVR）分类器的时间复杂度更高，因此后者也更加适用于大规模图像分类任务[95]。除了平分类器（flat classifiers，即所有分类一次性完成的分类器）以外，层次化的分类结构（可以经过多次分类达到目的）也经常被用于大规模的分类任务[96][97]。最后，选择合适的核函数（kernel functions）也对基于核方法（kernel methods）的分类器（如SVM）至关重要。尽管非线性核函数如χ^2函数[98]和Hellinger函数[7]都能提高分类精度，线性核函数被证明具有最好的可扩展性[99][100]以及较好的泛化性能。

2.2.5 特征索引和图像检索

大规模图像检索问题通常需要在短时间内查询局部特征的最近邻或者近似最近邻特征。倒排表（Inverted Index）[3][6]作为一种高效的数据结构被广泛应用。本质上说，倒排表是稀疏邻接矩阵的一种紧凑的存储方式，其中矩阵的每一行/列分别对应于一个局部特征/一张图像。倒排表以局部特征为索引，每一个局部特征都对应一个链表，存储着它出现的图像ID。为了后处理方便，其他有用的线索，如几何信息（特征在图像上的大小、位置等）也可以存储在倒排表中。在线检索过程需要访问并检查与查询图像共享至少一个局部特征的图像。由于倒排表的引入，减少了需要枚举的图像数量，从而使检索过程大大加速。

基于向量量化和标量量化，倒排表的构造过程有很大的不同。向量量化的结果是一个或多个经过加权的图像编号[8]，它们直接对应于倒排表的某个入口。然而，标量量化的结果，一个D'维的二值向量，却无法直接用于建立索引。此时，可以抽取这个二值向量的前t位（$t \ll D'$），将它的哈希值作为倒排表的入口（entry）。该向量的剩余部分（$D'-t$位）则作为额外信息被保存在倒排表里。这种方式不需要显式地训练视觉码本，而将二值化向量的前t位作为天然的视觉单词。此时，视觉单词的最大可能数量为2^t，在$t=32$[83]时可能多达4G。然而实际存在的单词并没有这么多：在大约1M（1百万）张图像构成的数据集上，码本的大小通常不超过100M（1亿），系统足以为每个视觉单词分配存储空间。

给定查询图像后，视觉词袋模型的前半部分（特征抽取和量化）也同样作用于查询图像，从而得到一个查询特征集合。这些特征被用于查询倒排表，以得到相应的相关候选图像列表，按照候选图像与查询图像的相关度（共享特征数目）排序。有时，我们也会对视觉单词进行简单的加权，以改善检索效果[101][102]。在标量量化模型中，在线查询的过程也发生了变化：此时的查询特征也是一些D'维的二值向量。这些向量的前t位被取出，计算哈希值，并且用于访问倒排表结构。额外存储的$D'-t$位向量可以用于检查该实例是否确实与查询特征相匹配。在实际应用中[83]，定义两个阈值：*编码扩展阈值*（codeword expansion threshold）d和*海明阈值*（Hamming threshold）κ。只有那些前t位与查询向量相异不超过d位的单词被访问，而全部D'位相异不超过κ位的最后被认定为匹配。给定t和d，需要访问的倒排表入口（单词）数为：$1+d+\frac{d(d-1)}{2}+\ldots+\binom{t}{d}$。$t$通常是固定的，而增加$d$则会带来召回率的提高和时间复杂度的上升。

为了提高检索效果，初始的候选图像排序结果通常需要进行*重排序*（re-ranking）。这个过程统称为*后处理*（post-processing），是提升检索精度的重要手段。最常见的后处理算法包括查询扩展、空间验证和扩散算法。*查询扩展*（query

expansion）[23][103][104] 通过重新考虑原先排名靠前的候选，将其中的特征加入查询范围，达到提升召回率的效果。空间验证（spatial verification）[105][6] 将那些不满足空间位置关系的特征匹配筛除，从而提升检索的准确率。空间验证还可以通过空间编码（spatial coding）进行[75][106][107]，或者通过构造视觉短语（visual phrases）从而将匹配单元扩大为更加鲁棒的特征组（feature groups）。扩散算法（diffusion-based algorithms）基于随机游走理论[108][24][109][110]，在图像和/或特征构成的图结构上传播置信值（affinity value），以获得候选图像的最后得分。除了这些典型方法外，还有大量其他的后处理算法，如选择高质量特征[111]、检测特征的共现性[112]、提取上下文信息[113][114][115]、利用最近邻特征信息[116][117]、利用其它匹配核[118]、融合多种特征[119]、考虑特征间相似性[120]，等等[121]。这些方法都能够有效地改善图像检索质量。

2.3 卷积神经网络

卷积神经网络（Convolutional Neural Network，CNN）是一种特殊的多层神经网络[122]，通过将图像信号的向前（正向）传递和向后（反向）传递，来修正网络的权值，以达到适应和学习的目的[123]。卷积神经网络可以看作多层感知器模型的一个变种：通过将其中的全连接层改为卷积层，可以显著地减少参数数量。从本质上看，卷积神经网络也可以视为一个复合函数，通过梯度计算的链式法则向后传递错误信号。当网络的层数以及每层的神经元足够多时，网络有能力拟合任意复杂的视觉概念。然而，训练所需的样本数量通常也随着网络的复杂性而增加，因此如何防止过拟合也成为一个重要的课题。

2.3.1 总体结构

最初的卷积神经网络是针对分类问题，尤其是图像分类问题而设计的[14]。对于具有C个类别的分类问题，有监督的输出信号通常服从C选1编码准则（1-of-C coding scheme），即预期输出信号为一个C维向量，其中只有第c维（c是标注的类别）等于1，其余都等于0。

卷积神经网络的一个重要单元是层（layers）。网络的每一层都由若干个神经元（neurons）构成。神经元可以具有多种类型，如卷积（convolution）神经元、池化（pooling）神经元、归一化（normalization）神经元、激励（activation）神经元，等等。不同类型的神经元可以对输入信号进行不同的处理，从而完成对应于视觉词袋模型的编码、组合等操作。一般来说，每层的神经元具有相同或相似的功能，而某些类型的神经元所具有的功能被证实与人类大脑对视觉信号的处理

相对应[57]。

通常，卷积神经网络的*输入层*（input layer）是图像信号，每个神经元对应图像上的一个像素点；其*输出层*（output layer）是任务信息（如分类结果）的某种编码，如常见的*C*选1编码准则。输入层和输出层中间的网络称为*隐藏层*（hidden layers），通常由具有特定功能的神经元组成，不同层的神经元可以具有不同的功能。常见的隐藏层包括：

- *卷积层*（convolution layers）通过对图像的卷积操作，获取某一局部的信息。与图像进行卷积的是*卷积核*（convolution kernels），其作用相当于视觉词袋模型中的码本。卷积核的个数决定了该卷积层的表达能力，不同卷积层的输出被视为不同的*信道*（channels）。卷积运算通常是卷积神经网络中最耗时的部分，也是加速算法重点考虑的部分（见第2.3.4节）。
- *池化层*（pooling layers）通过池化操作，对图像的局部信息进行综合。池化层的作用与视觉词袋模型中的池化操作非常类似。池化操作可以有很多种：除了最常见的*最大池化*（max-pooling）和*平均池化*（average-pooling）外，还可以有*随机池化*（stochastic pooling）[124]和*空间金字塔池化*（spatial pyramid pooling）[125]等。
- *归一化层*（normalization layers）对前馈信号进行归一化，从而尽量消除数据尺度对结果的影响。归一化可以在同一信道内进行，也可以在一定范围的相邻信道内进行。
- *激励函数层*（activation layers）利用激励函数对信号进行标准化。常见的激励函数包括*S形函数*（sigmoid function）和*修正线性单元*（Rectified Linear Unit，ReLU）。
- *损失函数层*（loss layers）通过计算当前输出信号与监督信号之间的差值，产生*错误信号*（error signal）并向后传递。常见的损失函数包括*绝对值差函数*和*softmax函数*。

从本质上看，卷积神经网络可以视为另外一种基于局部特征的图像表示模型。在较低的层次上，每个神经元只能够获得较小范围的图像输入，通常用于描述较弱的纹理、边缘等信息。随着卷积、池化等操作的进行，高层次神经元的控制区域也不断增加。在最后一层，神经元的控制区域通常能够覆盖整张输入图像，从而对全局特征进行描述。一个神经元能够接受输入信息的范围，称为这个神经元的*感受野*（receptive field）。

2.3.2 网络训练

卷积神经网络的训练过程就是向前和向后的信号传递过程。输入信号向前传

递，得到当前输入信号的估计输出，并且与监督信号对比，产生错误函数。随后，错误函数以梯度的形式向后传递，并且利用*梯度下降法*（gradient descent method）修改网络的参数，从而达到训练的目的。

由于单张图像对训练过程的影响比较随机，训练过程多采用*批次*（batch）处理的方式，即每次向前传递一批图像的信号，再在向后传递时对这些图像的错误信号进行平均以修正网络参数。批次训练方式是导致卷积神经网络消耗内存的主要原因，批量的大小和硬件条件紧密相关。为了达到更好的训练效果，训练过程需要在同一数据集上反复进行。每训练一次完整的数据集，称为网络经历了一代（epoch）。训练一个网络通常需要上百代。

当卷积神经网络结构的深度不断增加时，网络的参数不断增加，拟合能力也不断增强。如果训练数据的个数太少，就容易产生*过拟合*（over-fitting）现象。缓解过拟合的方法有很多种，例如通过*数据扩充*（data augmentation）增加训练数据量[126][11]，或通过对多个网络的输出进行平均[11][127][128]，又或在训练过程中加入随机噪声[129][130]，等等。为了加速训练，还可以在网络中间层加入错误信号，以深入监督训练过程[131]。其他一些技术也有助于训练更加强大的神经网络[132][133]。

2.3.3　其他应用

除了应用于图像分类任务，卷积神经网络还可以处理许多相关的问题。理论上说，只要能够利用监督信息定义损失函数，就可以利用卷积神经网络来优化这个函数，从而达到训练的目的。近年来，卷积神经网络已经运用到许多不同的计算机视觉问题中，如图像检索[134]、图像分割[135]、物体检测[12]，等等。在一些其他的研究领域，如翻译系统[136]和语音处理[137]，应用卷积神经网络或者其变种也能够有效地提高准确率。

从方法论上看，卷积神经网络的扩展方式主要包括两种：直接运用神经网络的中间输出结果作为区域特征，或者修改监督信息以适应其他问题。

2.3.3.1　深度特征

在利用分类监督信息训练卷积神经网络的过程中，卷积神经网络的权值不断被调整，直到适应相应的分类问题。经过大量数据训练后，卷积神经网络的中间输出也将具有一定的视觉含义[138]。因此，即便新的输入图像并不属于训练中的任何一类，卷积神经网络的中间输出结果也可以用于对图片进行语义描述。这种利用事先训练的网络提取的图片特征，通常被称为*深度特征*（deep features）。

提取深度特征可以被当作*迁移学习*（Transfer Learning，TL）[139]的一种简单有效的解决方案。事实证明[138][13]，卷积神经网络在大规模图像分类数据库上训练过后，其生成的深度特征具有很强的判别力，能够在一般的分类问题上产生良好的分类效果，即使新的分类并未出现在原先的数据集中。通常情况下，卷积神经网络在大规模分类问题上的准确率越高，迁移后的分类准确率也越高[140]。此外，挑选合适的层作为中间输出也有助于提高分类精度[13][140]。一般来说，更接近底层（输入）的输出信号具有更强的通用性，而接近高层（输出）的信号则更容易被特定的训练数据所拟合。当迁移前后的数据具有很强的相关性时，高层输出信号具有更强的描述能力；反之亦然[134]。

另外，深度特征也可以应用于其他计算机视觉任务。由于深度特征是一种高效的图像表示，它可以替代传统的图像特征，尤其是全局特征，以达到更好的精度。例如，对于近似近邻图像检索问题，可以利用深度特征描述图像，随后按照特征空间中的距离对候选图像进行排序。上述简单的方法即可达到与复杂算法（如局部特征配合后处理）相当的准确率[141][134]。此外，深度特征还可被运用于物体检测任务中，以判断提出的标注框确实包含物体的可能性[12]。

2.3.3.2 其他监督模式

对于某些任务，如图像分割和语音识别，直接利用深度特征进行处理是非常困难的。此时，通常将问题转化为一种新的目标函数，并利用卷积神经网络进行训练。对于图像分割，一个简单的思路是利用区域图像信息来判断中心像素是否为前景[49]。这是一个二分类问题，可以通过将卷积神经网络的输出层设置为两类来实现[135]。如果把二分类扩展为多分类，同样的方法就可以用于图像语法分析[49][142]。类似的思路也可用于文本理解或者语音识别，例如利用周围若干文本单词来表示中间的单词，从而进行压缩编码[143]。

2.3.4 网络的快速计算

卷积神经网络的一个重要特点是需要进行大量的算术操作。以2012年提出的针对ImageNet比赛设计的网络[11]为例，单线程计算需要超过6000小时。

繁重的计算任务主要来源于卷积运算（其本质为大矩阵乘法）。为了提高运算速度，可以利用缓存优化来加速矩阵乘法。其中一个常用的工具是*基础线性代数子程序库*（Basic Linear Algebra Subroutines，BLAS）①。根据实际测试的结果，通过快速缓存管理和汇编语言改写实现的BLAS快速矩阵乘法，比简单直接的矩

① http://www.netlib.org/blas/

阵乘法实现的运算速度快10倍以上。

另外一个加速手段是利用并行技术。包括卷积、池化、归一化等在内的多数神经元计算都可以分配给多个处理器同步执行。相比于中央处理单元（Central Processing Unit，CPU），图形处理单元（Graphics Processing Unit，GPU）具有更多的流处理器（stream processors），特别适合并行执行简单的计算任务（如矩阵乘法、像素级的池化和归一化等）。当前，开发GPU并行程序的主要方式是利用NVIDIA公司提供的CUDA®程序库①。基于CUDA实现的BLAS，即CUBLAS②，也能够有效加速基于GPU的矩阵乘法。

一般来说，GPU的内存（显存）比计算机的主内存要小得多。在某些情况下，可以通过多个GPU共同存储网络和中间数据[11]，以训练更大的网络模型。多个GPU的协同计算也能够有效地加速卷积神经网络的计算过程。

2.4 其他知识

最后，我们介绍一些本文中将会用到的相关技术，包括图像分割、边缘检测、物体检测和最近邻搜索。

2.4.1 图像分割

经典前背景分割问题的目标是找到一条闭合曲线，使得曲线的两边分别对应于图像的前景和背景。一般认为，两部分的边界应当具有较强的亮度差异，因此像素之间的亮度差值就可以作为该位置是否成为边界的加权因素。如果将每个像素看成一个节点（node），并且在相邻像素之间连上边（edge），整张图像就形成一个图（graph）结构。寻找最优的前背景分割，就是寻找这个图结构的最小割（minimum cut）。利用最大流最小割定理（max-flow-min-cut theorem），可以通过计算图的最大流（maximum flow）获得图像分割。这种算法成为一些常见算法的理论基础，如图切割算法（graph cut algorithm）[45]、抓取切割算法（Grab-Cut algorithm）[46]、绘画选择（Paint Selection）[144]，等等。其中，后两种算法还允许用户提供一定的标注信息以辅助切割，得到了广泛的应用。

随着深度学习的兴起，一种全新的思路被用于解决图像分割问题[135]。这种方法的思想更为简单：将图像分割看成逐像素的二分类问题，而分类所依赖的信息就是该像素周围一定尺寸的局部图像。由于训练样本数量大（一张图像里的每一个像素都可以看成一个独立的训练样本），可以运用深层卷积神经网络来拟合

① http://www.nvidia.com/object/cuda_home_new.html
② https://developer.nvidia.com/cuBLAS/

分类模型。此方法的优点是不需要人工先验信息的辅助，适合一般化大批量的训练。利用在大规模图像分类问题中训练的网络结构，能够快速处理测试图像，并且通过一定的后处理，达到较好的准确率。

2.4.2 边缘检测

边缘检测的评测标准是像素级别的：即需要对图像的每个像素，分别判断它是否属于边缘的一部分。由于图像中边缘的定义并不是绝对的，因此许多边缘检测的数据集采用了多人标注的方式以提供更好的测试环境[52]。

大部分边缘检测算法与计算图像亮度的梯度有关。典型的例子包括一阶的*Canny*算子（Canny operators）[145]、*Sobel*算子（Sobel operators）[146]、*Prewitt*算子（Prewitt operators）[147]，以及二阶的*Marr-Hildreth*算子（Marr-Hildreth operators）[148]。其中，Canny算子及其变种（包括Compass算法[149]等）是最为常用的边缘检测算法。其基本思想为：在降噪声（例如经过平滑高斯核卷积）的图像上计算亮度的梯度，并且利用最大抑制方法筛除一些假边缘。在后处理方面，利用高/低两种阈值筛除较弱的边缘响应，并且抑制那些出现在强响应附近，却不与强响应相连通的弱响应。如果在边缘响应上进行层次化的切割，还能够得到具有更强语义信息的边界响应[53]。

与图像分割类似，深度学习算法也能够用于边缘检测[150]。其基本性质与图像分割十分类似：需要决策神经元具有较大的感受野，且提供较大的训练数据集以防止过拟合。

2.4.3 物体检测

物体检测的基本问题设定是从给定图像上找出物体所在的位置，通常由一个框（bounding box）来表示。因此，大部分物体检测算法都与*滑动窗口*（sliding window）有关。

当前，对于特定物体，尤其是常见刚性物体的检测已经达到了令人满意的水平，如人脸检测（Face Detection）和行人检测（Pedestrian Detection）。由于人脸和行人具有较好的刚性，基于手工特征的滤波器（filters）通常能够得到较好的效果，如Gabor滤波器[151]和有向梯度直方图（Histogram of Oriented Gradients，HOG）[5]；基于简单特征和boosting的算法也能得到很好的结果[152]。

为了检测具有一定结构且能够变形的一般物体，如具有部位（part）概念的大部分动植物，*可变形的部位模型*（Deformable Part Model，DPM）[41] 同时优化部位匹配的外观精度和部位之间的几何位置关系。在著名的PascalVOC检测任

务[①]中，基于DPM的模型[41][153]在一段时间内都取得了领先的检测结果。改进的DPM[154]对于遮挡、变形等常见噪声都具有较强的鲁棒性。然而也有学者指出：DPM受限于所采用滤波器的描述能力[155]，无法达到令人满意的检测水平。

近年来，一种新的思路被应用于物体检测[12]：先在图像上抽取大量的候选物体区域，再从中筛选出最有可能的答案。提取候选物体区域的方法被称为*物体性*（Objectness）判断[156]。常见的方法有*选择性搜索*（Selective Search）[157]和基于压缩编码的二分类算法（如*BING算法*[158]）。评价物体性判断的主要标准是召回率。在每张图片上提出大约1000个候选区域时，上述方法基本都能达到90%以上的召回率。最后，我们通过对候选区域的筛选，找出一个或少数几个真正的物体。基于深度神经网络的图像分类技术能够在筛选环节上取得更好的效果[12][159]。近年来，在ImageNet大型物体检测竞赛[②]中，上述模型已经超过基于DPM的算法，处于显著的领先地位。

2.4.4 最近邻搜索

最近邻搜索问题的基本设定如下：给定一个D维向量$\mathbf{q}$，以及由N个D维向量构成的数据库$\mathcal{X} = \mathbf{x}_1, \ldots, \mathbf{x}_N$，需要寻找$\mathcal{X}$中与$\mathbf{q}$距离（采用某种度量，例如欧氏距离）最近的向量。最简单的最近邻搜索算法称为*暴力搜索*（bruteforce search），它枚举所有的候选向量，逐一计算其距离并找出最小的一个。计算两个D维向量之间距离的典型复杂度为$O(D)$，因此暴力搜索的总时间复杂度为$O(ND)$。在大规模搜索中，暴力搜索的主要缺点是查询时间太长。例如，常用于最近邻搜索算法评价的SIFT1B数据库[160]中包含有10^9个128维向量。暴力搜索通常需要超过20秒的时间（单核3.0GHz处理器）才能返回一个查询向量的结果。

快速的最近邻搜索算法通常通过牺牲一定的精度来达到加速的目的。一个典型的例子是*近似近邻搜索*（Approximate Nearest Neighbor，ANN）[161]，能够对于任意$\epsilon > 0$，保证以$1 - \epsilon$概率得到最近邻搜索结果[162][163]。利用某些为高维情形设计的数据结构如*k维树*（*k*d-tree）[164]，我们能够在搜索精度和复杂度上达到很好的平衡。由此产生的近似算法，如FLANN[165]，被应用于其他大规模计算，如近似近邻聚类问题[6]。

另一类常见的近似近邻搜索算法，将高维向量映射到低维子空间或者进行压缩编码，同时最大限度地保留原空间的几何特性。一个典型的例子是基于*哈希*（Hashing）的算法。哈希法，又称散列法，将高维向量通过若干布尔函数映射为紧凑编码[166]。其中，*局部敏感哈希*（Locality-Sensitive Hashing，LSH）算

① http://pascallin.ecs.soton.ac.uk/challenges/VOC/

② http://www.image-net.org/challenges/LSVRC/

法[167]通过定义若干超平面来进行二值化，并在二值化的编码上逐一枚举并计算海明距离（Hamming distance），使得计算速度大幅提升。哈希法的缺点是容易产生大量距离相同的编码，此时由于原始信息已经丢失，无法判断其先后顺序。

另外一种向量编码的算法称为乘积量化（Product Quantization，PQ）[168]。PQ将K维向量切分为M段（通常M能整除D），并且在每一段构建若干规模较小的码本（codebook）。每个向量被压缩成M个整数，每个整数表示在相应段的码本中最近邻向量的编号。在线查询时，可以先进行一次预处理，计算查询向量的每一段到相应码本中每个向量的距离，然后枚举并计算查询向量$\mathbf{q}$到每个候选向量$\mathbf{x}_n$的距离。由于在线部分只需要进行M次加法计算，因此能够快速完成（对比于D次乘法计算）。 PQ算法的改进，如最优乘积量化（Optimized PQ，OPQ）[169]， 补偿乘积量化（Composite PQ，CPQ）[170]等，都得到了广泛的应用。基于PQ的方法在SIFT1B数据库上的查询时间通常不超过1秒（单核3.0GHz处理器）。

第 3 章　局部特征的翻转不变强化

3.1　研究动机

在图像的分类和检索任务中，局部特征的鲁棒性和*不变性*（invariance）至关重要（见第2.2.1节）。传统的DoG[4]、MSER[62]、Hessian Affine[61]等检测子，以及SIFT[4]、HOG[5]、LCS[7]等描述子，利用对图像上*感兴趣区域*（Regions-of-Interest，RoI）的*主方向*（dominant orientation）和*尺度*（scale）进行估计，并且对局部区域的几何特性进行相应调整，使得局部特征具有*尺度不变性*（scale invariance）和*旋转不变性*（rotation invariance）。

然而，上述局部特征并不具有*翻转不变性*（reversal invariance）。这就意味着，在一张图像被翻转后，其中提取的局部特征可能会与原先完全不同，从而很难在原图和翻转图像中建立特征对应关系。相应地，计算而得的图像特征表示就会完全不同，从而极大地影响分类和检索算法的精度，如图3.1所示。为了应对这种不稳定性，研究者们通常利用*数据扩张*（data augmentation）的方法，将数据集里的每一张图像都进行翻转，同时使用翻转前和翻转后的图像进行训练和测试，以提供更多的数据[171][90]。虽然这种方式有效提高了识别精度，但它们带来的额外时间和空间开销仍然不可忽视。

本章主要讨论如何使得局部特征（描述子）具有翻转不变性。我们将从观察入手，分析SIFT特征在翻转后的变化，并且提出一种基于字典序比较的简单算法Max-SIFT，消除翻转带来的影响。随后，我们将Max-SIFT扩展为更一般的RIDE算法，使它能够以SIFT朝向为基础，为其他局部特征提供翻转不变特性。我们的算法已经在大量图像分类数据集上进行了测试，都取得了很好的效果。

与本章相关的出版物为[172][173]。

3.2　翻转不变性的重要性

本节通过几个例子，进一步论述局部特征的翻转不变性在图像分类和检索问题中的重要性。

在几乎所有*细粒度*（fine-grained）图像分类数据集中，都同时存在着*朝向*（orientation）不同的物体。例如，在包含11788张图片的Caltech-UCSD-**Bird-200**-2011数据集[34]中，分别存在至少5000张明显朝左和5000张明显朝右的图片；在

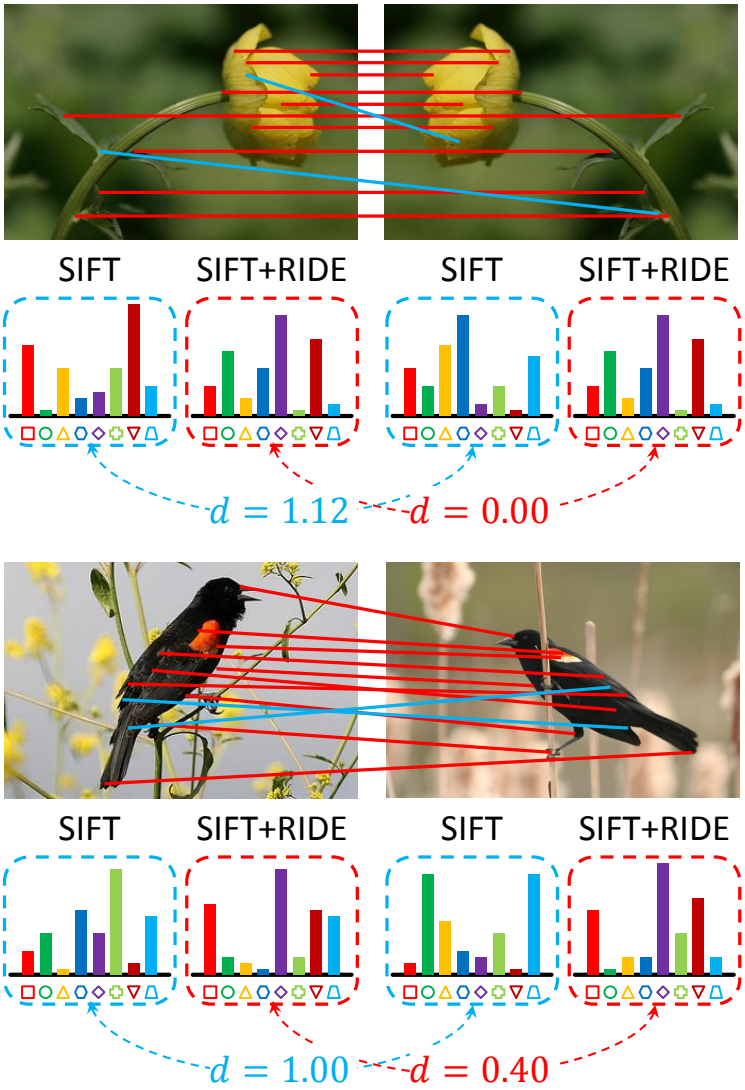


图 3.1　SIFT特征[4]在使用RIDE算法（红色）和不使用RIDE算法（蓝色）时的特征匹配结果和相应图像表示。在不使用RIDE技术时，即使在一张图和它的翻转副本之间，都很难找到有效的特征匹配。 RIDE还显著地缩小了翻转物体/图像之间的距离（视觉词袋特征）。

包含10000张图像的**Aircraft-100**数据集[36]中，则有至少4800张图像朝左，另外至少4500张图像朝右。

这种朝向的差别对于分类问题的影响是很大的。我们用一个在**Aircraft-100**数据集上的简单实验来说明这个问题。之所以选择这个对飞机图像进行分类的数据集，是因为飞机的朝向比鸟类更容易判断。基于原始数据集，我们手工将那些（机头）朝左的图像调整为朝右（进行翻转），以生成一个全部朝右的数据集。在这个经过左右朝向对齐数据集上，我们提取图像的视觉词袋特征（见第3.4.1节），分别进行分类和检索实验。

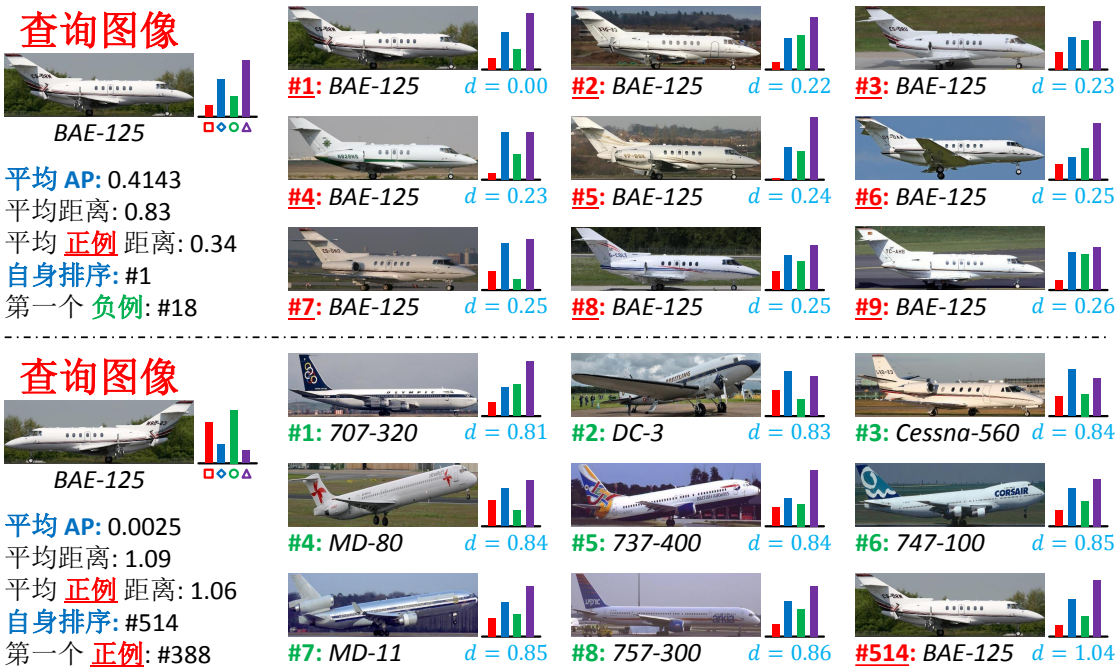


图 3.2　使用传统视觉词袋模型抽取的特征，在一个所有图像都朝右的数据库上进行检索：利用一张朝右的图像和一张内容相同但朝左的图像进行查询，将得到完全不同的检索结果。

我们利用该数据集的标准分类设置（大约2/3图像用于训练，其余用于测试）。在原始数据集（未对齐）上，得到53.13%的分类准确率；而在对齐后的数据集上，准确率迅速提升至63.94%（相对提升超过20%）。这说明，朝向的对齐给细粒度分类问题带来了巨大的好处。反之，我们将所有的10000张对齐的图像（全部朝向右）都用于分类训练，并且将这些图像的翻转副本（10000张朝左图像）用于测试。此时，训练集和测试集为完全相同的图像样本，测试分类准确率应当接近100%，然而实际准确率只有46.48%，远远低于预期。这表明，一个在朝右物体上训练而得的分类模型，在预测朝左物体的类别时，并不具有很好的泛化性能。

为了获得一种直观的解释，我们在朝右对齐的数据集上进行检索任务。一个典型的查询图像和相应的按照欧氏距离（差向量的ℓ_2范数）的检索结果如图3.2所示。当查询图像具有和数据集整体相同的朝向（朝右）时，检索结果是基本令人满意的： mAP值为0.4143，且第一个错误的样例出现在第18位（前17位结果都属于正例）。然而，当查询图像被翻转过来（朝左）时，图像上的所有特征都相应翻转，从而导致图像的特征表示被完全改变。此时，检索结果大幅恶化：mAP值为0.0025，第一个正例出现在388位（前387位结果都属于负例）。值得注意的是，在后一种情况下，翻转后查询图像本身出现在结果的第514位，这意味着在全部10000张图像中，有超过500张图像比翻转后的样例更接近查询图像!

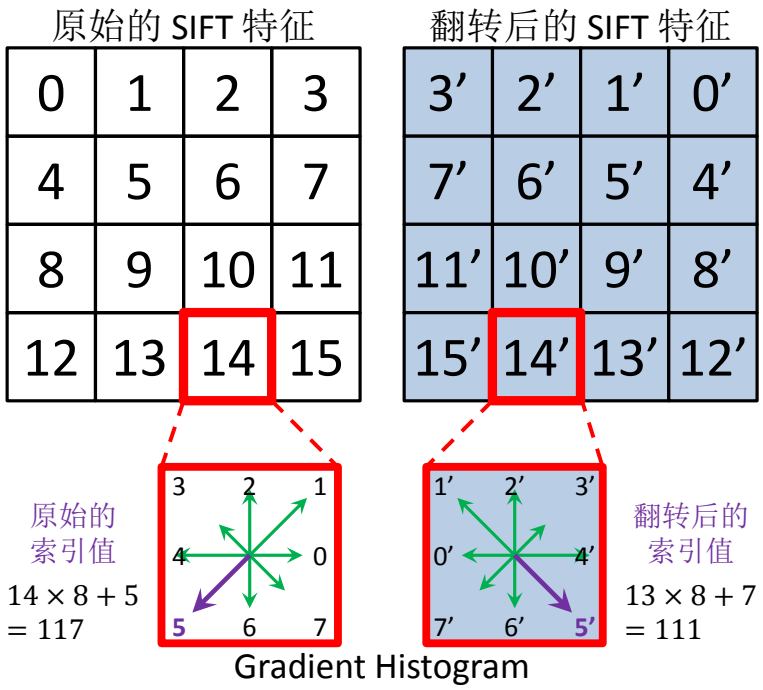


图 3.3　一个SIFT特征和它翻转后的特征。同样的数字表示对应的网格/梯度方向。原SIFT特征中标注的数字表明了网格枚举和梯度收集的顺序。

正是由于一张图像和它的翻转副本之间具有如此迥异的特征表示，在细粒度分类数据集中，我们事实上需要针对同一类视觉语义，处理两种甚至更多的物体原型（prototypes）。因此，原本就有限的训练数据被分割为更多部分，导致每个原型能够得到的训练数据都显著地减少了。基于这种考虑，一些算法[171][90]采用了数据扩张的方式来增加每个原型的训练样例数。我们采用一种不同的思路：不增加图片数量，而是针对局部描述子进行处理，使得它们在翻转操作中具有不变性。

3.3　翻转不变的局部特征

本节为本章的主要部分。我们首先讨论局部特征在翻转后的性质变化，随后提出两种基于对称操作的算法以抵消翻转操作。最后，我们将翻转不变的特征应用于图像分类任务上，并讨论我们的方法与已有算法的区别。

3.3.1　局部特征的翻转

SIFT是一种基于空间分划和朝向直方图的局部特征，其结构如图3.3所示。一个局部图像块被分割为4 × 4网格。在每个网格里，我们计算一个8维的梯度直方图向量。这里，我们假设空间网格从上到下、从左到右枚举，而梯度强度按照逆

时针方向收集。当图像被左右翻转后，其中所有的小块也随之左右翻转，但网格的枚举顺序和梯度的收集顺序依然保持不变，由此造成每个网格的实际出现顺序改变，网格内的梯度收集顺序也从$(0,1,2,3,4,5,6,7)$变为$(4,3,2,1,0,7,6,5)$（见图3.3），这正是SIFT特征不满足翻转不变性的原因。

记原始的SIFT特征（128维）为$\mathbf{d} = (d_0, d_1, \ldots, d_{127})$。其中，对所有的$i = 0,1,\ldots,15$和$j = 0,1,\ldots,7$，有$d_{i\times 8+j} = a_{i,j}$。如图3.3所示，原SIFT特征中的每个维度（0到127）都被一一映射到翻转后SIFT特征的某个维度。以图3.3中14号网格的第5个梯度方向（在图中以红色粗箭头表示）为例，它在原特征中位于第$14\times 8+5 = 117$维，在翻转后，同样的梯度强度出现在第13号网格的第7个梯度方向，在特征中位于第$13\times 8+7 = 111$维。我们定义映射函数$f^{\mathrm{R}}(\cdot)$（即$f^{\mathrm{R}}(117) = 111$），则翻转后的SIFT可以表示为：$\mathbf{d}^{\mathrm{R}} = f^{\mathrm{R}}(\mathbf{d}) = \left(d_{f^{\mathrm{R}}(0)}, d_{f^{\mathrm{R}}(1)}, \ldots, d_{f^{\mathrm{R}}(127)}\right)$。显然，$\left(\mathbf{d}^{\mathrm{R}}\right)^{\mathrm{R}} = \mathbf{d}$：这意味着连续两次翻转一个特征，将得到原始特征。

3.3.2　Max-SIFT特征

为了得到翻转不变性，我们需要定义一种特征转换函数：$\widetilde{\mathbf{d}} = r(\mathbf{d}) = s\left(\mathbf{d}, \mathbf{d}^{\mathrm{R}}\right)$，其中对于任意的特征对$(\mathbf{d}_1, \mathbf{d}_2)$，$s(\cdot,\cdot)$满足对称性，即$s(\mathbf{d}_1, \mathbf{d}_2) = s(\mathbf{d}_2, \mathbf{d}_1)$。这样就可以保证翻转不变的特性：$r(\mathbf{d}) = s\left(\mathbf{d}, \mathbf{d}^{\mathrm{R}}\right) = s\left(\mathbf{d}^{\mathrm{R}}, \mathbf{d}\right) = s\left(\mathbf{d}^{\mathrm{R}}, \left(\mathbf{d}^{\mathrm{R}}\right)^{\mathrm{R}}\right) = r\left(\mathbf{d}^{\mathrm{R}}\right)$。

许多对称函数都可以作为$s(\cdot,\cdot)$的候选，包括逐维求和或者逐维求最大/最小值。为了保证描述子的描述性能得到最大程度的保留，我们只选择这样一类特殊的函数：在任何情况下，$s\left(\mathbf{d}, \mathbf{d}^{\mathrm{R}}\right)$等于$\mathbf{d}$和$\mathbf{d}^{\mathrm{R}}$中的某一个。一般地，我们定义一个量化函数$q(\cdot)$，并依此选择$\mathbf{d}$和$\mathbf{d}^{\mathrm{R}}$中量化函数值较大的一个。

一种最简单的算法称为Max-SIFT。它直接将量化函数定义为特征第0维的梯度强度值：$q(\mathbf{d}) = d_0$。也就是说，直接对比每个图像块的翻转前后的SIFT特征的第1维，将较大者认定为描述该图像块的特征。如果第0维相等（即原始特征的0维和第28维相等，因为$f^{\mathrm{R}}(0) = 28$），那么再比较翻转前后SIFT特征的第1维、第二维、……，依此类推，直到分出大小为止。在SIFT特征$\mathbf{d}$上计算的Max-SIFT特征可以记为$m(\mathbf{d})$。

3.3.3　RIDE算法

上述Max-SIFT算法直接将SIFT的第0维作为量化函数。SIFT的第0维指向正右方（见图3.3），因此可以认为第0维在某种程度上表达了这个SIFT特征的“朝向右方的程度”。这样，Max-SIFT算法实际上选择了翻转前后的SIFT特征中，更有可能朝向右方的一个作为最后的特征。将这种直观算法以一种更加一般化的形式

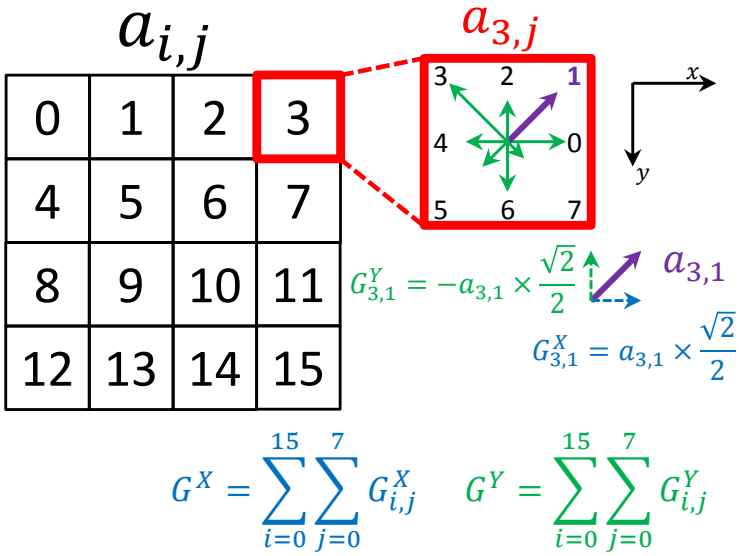


图 3.4　估计SIFT特征的总体主朝向的示意图。

表达出来，就成为下面介绍的**翻转不变性特征强化**（Reversal Invariant Descriptor Enhancement，RIDE）算法。

理想情况下，量化函数$q(\mathbf{d})$应该能够捕捉$\mathbf{d}$的朝向信息，即$q(\cdot)$反映局部特征的“朝向右方的程度”。原始版本的SIFT特征[4]本身就带有朝向信息（每个特征都有主朝向$\theta \in [0, 2\pi)$），只不过朝向信息在密集采样过程中被忽略了[64][174]。我们的目标就是通过密集采样的SIFT特征里每一维的梯度强度，恢复该SIFT特征的总体主朝向信息。

这里一个基本的结论是：SIFT特征的整体朝向可以通过其局部朝向估计出来。对于128维中的每一维，我们将它的梯度乘以它对应的单位朝向向量（8个方向之一），就得到其对应的局部朝向向量。这相当于将每一个梯度值按照x和y方向分解为两个值，其中x分量的值就对应于该梯度值所贡献的“朝右程度”。累积所有128维的x和y分量值，就得到了SIFT特征的整体朝向信息。关于这个结论的证明见附录A.1。现在，定义8个二维单位向量$\mathbf{u}_j$，$j = 0, 1, \ldots, 7$。根据SIFT特征的定义（图3.3），我们有：$\mathbf{u}_j = (\cos(j\pi/8), \sin(j\pi/8))^{\top}$。这样，整体朝向就可以写为：$\mathbf{G} = \left(G_x, G_y\right)^{\top} = \sum_{i=0}^{15}\sum_{j=0}^{7} a_{i,j}\mathbf{u}_j$。计算过程如图3.4所示。

我们可以简单地将G_x作为所需的量化函数值，即令$q(\mathbf{d}) = G_x(\mathbf{d})$对所有$\mathbf{d}$成立。值得注意的是，$q(\mathbf{d}) = -q\left(\mathbf{d}^{\mathrm{R}}\right)$对于任何$\mathbf{d}$都成立，因此我们可以利用符号$q(\mathbf{d})$来计

算法：推广的RIDE算法

1. 输入：
   一个原始局部特征的集合：$\mathcal{D} = \{\mathbf{d}_m, \mathbf{l}_m\}_{m=1}^{M}$。
2. 步骤：
   - 计算翻转特征集合：$\mathcal{D}^{\mathrm{R}} = \left\{\mathbf{d}_m^{\mathrm{R}}, \mathbf{l}_m\right\}_{m=1}^{M}$；
   - 如果需要，计算对应的SIFT特征集合：$\mathcal{D}^{\mathrm{S}} = \left\{\mathbf{d}_m^{\mathrm{S}}, \mathbf{l}_m\right\}_{m=1}^{M}$；
   - 利用SIFT特征计算朝向量化函数：$q(\mathbf{d}_m) = G_x\left(\mathbf{d}_m^{\mathrm{S}}\right)$；
   - 通过(3-1)式，计算翻转不变的特征：$\widetilde{\mathbf{d}_m} = r(\mathbf{d}_m)$。
3. 输出：
   一个翻转不变的局部特征集合：$\widetilde{\mathcal{D}} = \left\{\widetilde{\mathbf{d}_m}, \mathbf{l}_m\right\}_{m=1}^{M}$。

图 3.5　RIDE算法的一般化流程。

算一种翻转不变的特征变换$\widetilde{\mathbf{d}}$：

$$\widetilde{\mathbf{d}} = r(\mathbf{d}) = \begin{cases} \mathbf{d} & q(\mathbf{d}) > 0 \\ \mathbf{d}^{\mathrm{R}} & q(\mathbf{d}) < 0 \\ \max\left\{\mathbf{d}, \mathbf{d}^{\mathrm{R}}\right\} & q(\mathbf{d}) = 0 \end{cases} \tag{3-1}$$

其中，$\max\left\{\mathbf{d}, \mathbf{d}^{\mathrm{R}}\right\}$表示$\mathbf{d}$和$\mathbf{d}^{\mathrm{R}}$中具有较大字典序的一个。这意味着，当量化函数无法判断朝向信息时，我们简单地计算Max-SIFT描述子以达到翻转不变特性。我们将$r(\mathbf{d})$称为原特征$\mathbf{d}$经过RIDE处理后的特征。

3.3.4　将RIDE扩展到其他局部特征

此节，我们将RIDE扩展到其他局部特征，并且讨论更加一般化的翻转不变性质。

当RIDE算法应用于其他特征时，我们可以先在同样的区块上相应计算SIFT特征。随后，我们计算朝向向量$\mathbf{G} = \left(G_x, G_y\right)^{\top}$以（利用SIFT特征）估计该区块的朝向信息，并且在需要时进行翻转操作。图3.5展示了一般化的RIDE算法流程。 RIDE算法在特征变换中的额外时间复杂度主要来源于计算SIFT特征，而这个步骤只在需要时进行。例如，RGB-SIFT特征由三个信道（红绿蓝）的SIFT特征$\mathbf{d}_{\mathrm{R}}$、$\mathbf{d}_{\mathrm{G}}$和$\mathbf{d}_{\mathrm{B}}$组成，于是我们可以先分别计算每个信道的朝向向量$\mathbf{G}_{\mathrm{R}}$、$\mathbf{G}_{\mathrm{G}}$和$\mathbf{G}_{\mathrm{B}}$，然后利用RGB到灰度的变换来快速计算整体朝向向量：

$\mathbf{G} = 0.30\mathbf{G}_{\mathrm{R}} + 0.59\mathbf{G}_{\mathrm{G}} + 0.11\mathbf{G}_{\mathrm{B}}$。对于其他的颜色SIFT特征，我们也可以利用类似的方法，通过颜色信道的组合系数，重构灰度SIFT特征。

RIDE算法还可以用于抵消更大范围内的翻转操作，包括上下翻转以及图像的规则旋转（90°、180°和270°）。左右翻转、上下翻转以及图像旋转90°总共可以产生8种不同的特征描述子（见附录A.2.1节）。为了抵消这些变换，我们必须在朝向量化向量$\mathbf{G}$上施加更严格的限制。注意到限制$G_x > 0$可以从2个候选中找到1个，达到左右翻转不变性；类似地，同时限制$G_x > 0$和$G_y > 0$可以从4个候选中找到1个，同时达到左右和上下翻转不变性；同时限制$G_x > G_y > 0$从8个候选中找到1个，同时达到左右和上下翻转以及90°旋转不变性。我们将上述三种算法分别记为RIDE-2、RIDE-4和RIDE-8，其中的数字表示候选的特征数量。在附录A.2.2中，我们将论述：在不必要的情形下使用RIDE算法会带来性能的下降。对于一般分类数据集，只有左右翻转变化是常见情形，因此只有RIDE-2能够提升分类效果。因此，在接下来的章节中，我们所提到的RIDE算法都是指RIDE-2算法。

3.3.5　图像应用

我们简单地叙述Max-SIFT和RIDE算法在图像分类问题中的应用。考虑一张图像$\mathbf{I}$，以及一个局部特征（如SIFT）的集合：$\mathcal{D} = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_M\}$。当图像被左右翻转后，集合$\mathcal{D}$中的所有特征也被相应翻转，成为：$\mathcal{D}^{\mathrm{R}} = \left\{\mathbf{d}_1^{\mathrm{R}}, \mathbf{d}_2^{\mathrm{R}}, \ldots, \mathbf{d}_M^{\mathrm{R}}\right\}$。如果这些局部特征并不具有翻转不变特性，即$\mathcal{D} \neq \mathcal{D}^{\mathrm{R}}$，那么通过集合$\mathcal{D}$和$\mathcal{D}^{\mathrm{R}}$计算出的图像全局表示将完全不同。通过Max-SIFT或RIDE算法，我们可以保证对于任何特征$\mathbf{d}$都有$\widetilde{\mathbf{d}} = \widetilde{\mathbf{d}^{\mathrm{R}}}$，这样集合$\widetilde{\mathcal{D}}$和$\widetilde{\mathcal{D}^{\mathrm{R}}}$就完全相等，从而产生相同的图像表示。

当RIDE算法与特征的空间组合算法（如SPM[10]）配合使用时，有一个值得注意的小技巧。注意到左右翻转前后对应的描述子在图像中可能具有不同的x坐标。例如，一个出现在原图像左上角的特征，将会出现在翻转后的图像的右上角。如果直接使用SPM算法对图像进行切割和池化，就会导致这个特征被分配到不同的池化组中，从而造成翻转后图像表示的不一致。为了解决这个问题，我们计算一张图像中被Max-SIFT或者RIDE算法翻转的特征（$\widetilde{\mathbf{d}} \neq \mathbf{d}$）个数，并且与特征总个数进行对比，如果翻转的特征超过了总数的一半，就认为这张图像应该被整体翻转。此时，我们用$\mathcal{D}^{\mathrm{R}}$替代$\mathcal{D}$，包括其中的横坐标值（x被替换为$W - x$，其中W是图像宽度）。这个过程等价于利用局部特征的统计信息来预测图像的整体朝向。

3.3.6　与已有方法的对比

尽我们所知，Max-SIFT和RIDE算法虽然简单，但是在此前的工作中并没有得到广泛应用。许多近期发表的论文[9][171][90][175]都通过数据扩张来处理图像翻转带来的变化。我们将在后续实验部分说明，RIDE算法能够达到比数据扩张方法更好的效果，同时还具有更低的时间和空间复杂度。

虽然某些具有翻转不变特性的局部特征已经被用于图像检索任务[176][177][178][172]，但这些特征并没有在图像分类问题中得到关注。作为参照，我们也实现了MI-SIFT[177]，并且将它与Max-SIFT和RIDE算法进行对比。在实验中，RIDE算法产生的分类准确率显著地超过了Max-SIFT和MI-SIFT算法。这说明，RIDE算法比之前的算法更好地抓住了密集采样特征的一些特性（如朝向性质），从而与图像分类模型配合得更好。

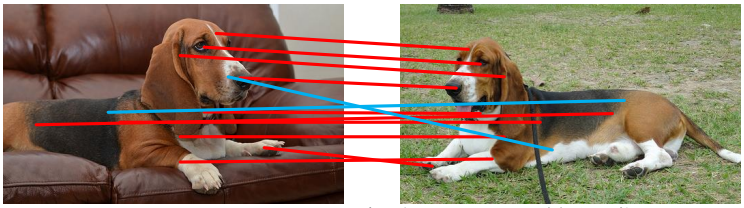
3.4　实验部分

本节包含大量的实验结果，用于展示RIDE算法为局部特征匹配、细粒度分类和场景分类问题带来的好处。

3.4.1　数据集和基本设置

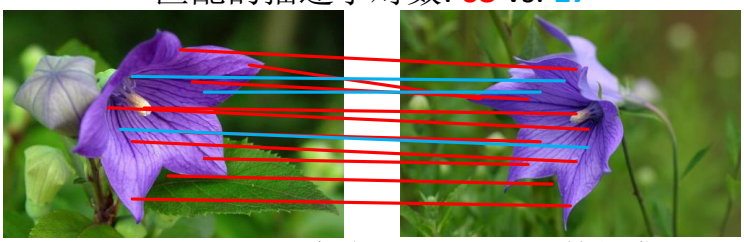
我们在四个公共的细粒度物体识别数据集上测试我们的算法： Oxford **Pet-37**数据集[179]（包含37种宠物猫或狗，7390张图像）； **Aircraft-100**数据集[36]（100种飞机模型，每个模型有100张图像）； Oxford **Flower-102**数据集[33]（分属102种花的8189张图像）；以及Caltech-UCSD **Bird-200**-2011数据集[34]（11788张鸟类图像，分属200个不同的物种）。在**Aircraft-100**和**Bird-200**数据集上，每张图像都提供了一个物体的包围框（bounding box）。这几个数据集都提供了标准的训练和测试样例的划分，每一类的训练图像个数（依次）大约为100、20、67和30。

基本的实验设定遵循近期提出的一个基于Fisher向量[7]的视觉词袋模型[19]。一张图像首先被重置大小，在保证其长宽比不变的情况下，将其长边重置为300个像素。如果该数据集还提供了物体的包围框，那么只有包围框内的图像被使用并且重置大小。我们用VLFeat[174]，一个通用的计算特征代码库，提取密集的RootSIFT特征[118]。密集采样的空间跨度为6像素，窗口大小为12像素。在相同的图像块中，我们还计算了LCS[7]、RGB-SIFT和Opponent-SIFT[70]特征。随后，RIDE算法被应用于每一个单独的特征上。 SIFT和LCS特征被PCA降维至64维，而基于颜色信道的SIFT特征被PCA降维至128维。我们利用具有32个分量的高斯混合模型（GMM）对每一种特征进行聚类，并且用改进的Fisher向量[7]进行编

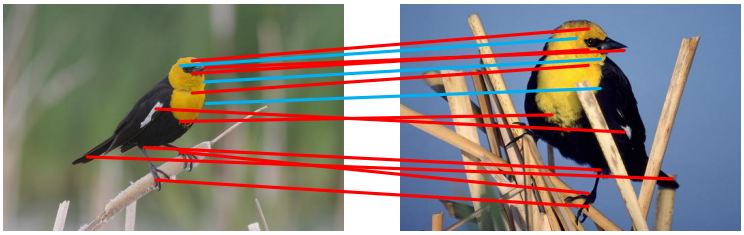


(a) *Basset Hound* 来自 **Pet-37** 数据集
匹配的描述子对数: **37** vs. **2**

(b) *Boeing 777-300* 来自 **Aircraft-100** 数据集
匹配的描述子对数: **63** vs. **17**



(c) *Flower#017* 来自 **Flower-102** 数据集
匹配的描述子对数: **41** vs. **6**



(d) *Yellow-hd Blackbird* 来自 **Bird-200** 数据集
匹配的描述子对数: **51** vs. **14**

图 3.6　利用原始SIFT（蓝色）和经过RIDE处理后（红色）的特征进行局部特征匹配的结果。图像上的连线，展示了不超过10个RIDE匹配和3个SIFT匹配的详细位置。

码。我们使用4个区域的空间金字塔模型（整幅图像和三个水平长条区域）对特征进行空间编码。 SIFT和LCS产生的特征向量可以进行拼接，以产生融合的（FUSED）特征向量。每一种特征向量先用平方根归一化再用ℓ_2归一化[180]，最后被送入LibLINEAR[99]，一个通用的线性SVM模型，以进行训练和测试。我们报告的分类准确率，是算法在所有类的测试图像上的平均分类精度。

为了将我们的结果和现有的先进结果进行比较，我们还提取了更为强大的高维特征。为此，我们将图像的大小重置为长边具有600像素，并且利用空间跨度8、窗口大小16的密集采样，以及具有256个分量的GMM模型。

3.4.2　局部特征的匹配

我们首先进行局部特征匹配实验。在每个数据集上，我们选择了一对具有不同朝向的图像（物体），在其上计算SIFT特征并用RIDE算法进行强化处理。我们对于每个可能的局部特征匹配计算欧氏距离，并利用一个固定的阈值来筛选匹配的特征对。随后，我们使用空间验证算法来筛除那些不恰当的匹配。

一些典型的匹配结果如图3.6所示。我们可以观察到，RIDE算法显著提升了特征匹配的数量和质量（大量原本无法匹配的对应点都能被找到）。这说明，RIDE特征有助于增强原本不具有翻转不变特性的SIFT特征，使我们能在不同朝向的物体上发现对应的位置和特征。

3.4.3　细粒度物体识别

我们在表3.1中报告不同特征和不同算法在细粒度物体识别上的实验结果。在原始特征（SIFT、LCS、颜色SIFT）上，我们采用了**RIDE**和数据扩张（**AUGM**）的方法，并且对比它们的效果。在这里，数据扩张的含义是：对于每张训练和测试图片，都产生一个额外的左右翻转副本。我们使用两倍于原有设定的图像集训练模型（SVM），在测试时同时测试翻转前后的样例，并利用softmax函数[175]融合结果，产生最后的输出。

在表3.1中，我们可以观察到**RIDE**算法在原始特征（**ORIG**）上产生了稳定的分类精度提升。此外，当我们使用SIFT和颜色SIFT特征时，**RIDE**算法甚至比数据扩张（**AUGM**）方法的分类精度更高。当我们使用LCS特征时，**RIDE**算法的分类精度略低于数据扩张（**AUGM**）方法，这可能是因为LCS特征（一种不基于梯度信息的特征）的朝向并不能被SIFT的梯度值很好地估计。

我们强调，由于训练的时间和空间复杂度大致与训练的数据量成正比，数据扩张（**AUGM**）方法几乎需要**RIDE**算法两倍的计算开销（参见第3.4.6节）为了在计算复杂度相当的情况下公平地对比两种算法，我们将**RIDE**算法中使用的码本大小翻倍。这样，就产生了一个时间和空间复杂度都和数据扩张（**AUGM**）方法相当的**RIDE**算法。我们记这种方法为**RIDE**×2，它能够稳定地比数据扩张（**AUGM**）方法产生更好的分类结果。

我们同样使用了强特征（见第3.4.1节），并且将结果与近期的工作进行对比，包括与我们的算法有很强联系的MI-SIFT[177]方法。在表3.2中，我们的方法展现出了很有竞争力的实验结果。此外，研究者们还使用一些复杂的部件检测器用于对诸如**Bird-200**这样的数据集进行分类，并且取得了非常好的实验结果[90][91][184][159]。为了让RIDE算法与部件检测方法配合，我们利用

	ORIG	**RIDE**	**AUGM**	**RIDE×2**
SIFT	37.92	42.28	42.24	**45.61**
LCS	43.25	44.27	45.12	**46.83**
FUSED	52.06	54.69	54.67	**57.51**
RGB-SIFT	44.90	47.35	46.98	**49.53**
OPP-SIFT	46.53	49.01	48.72	**51.19**

(a) **Pet-37**的分类结果

	ORIG	**RIDE**	**AUGM**	**RIDE×2**
SIFT	53.13	57.82	57.16	**60.14**
LCS	41.82	42.86	43.13	**44.81**
FUSED	57.36	61.27	60.59	**63.62**
RGB-SIFT	57.89	63.09	62.48	**65.11**
OPP-SIFT	47.06	53.12	51.39	**55.79**

(b) **Aircraft-100**的分类结果

	ORIG	**RIDE**	**AUGM**	**RIDE×2**
SIFT	53.68	59.12	58.01	**61.09**
LCS	73.47	75.30	75.88	**77.40**
FUSED	76.96	80.51	79.49	**82.14**
RGB-SIFT	71.52	74.97	74.18	**77.10**
OPP-SIFT	76.12	79.68	78.83	**81.69**

(c) **Flower-102**的分类结果

	ORIG	**RIDE**	**AUGM**	**RIDE×2**
SIFT	25.77	32.14	31.60	**34.07**
LCS	36.18	38.50	38.97	**40.16**
FUSED	38.11	44.73	43.98	**46.38**
RGB-SIFT	31.36	39.16	38.79	**41.73**
OPP-SIFT	35.40	42.18	41.72	**44.30**

(d) **Bird-200**的分类结果

表 3.1　不同特征产生的分类准确率（%）。这里，**ORIG**和**RIDE**分别表示不使用和使用RIDE算法处理的模型， **AUGM**表示在原始特征上采用数据扩张方法，而**RIDE×2**表示在RIDE处理后的特征上采用双倍的GMM码本。

了[90]和[91]发布的检测结果，并且在其上实现了原始SIFT特征和RIDE算法。实验结果表明，RIDE算法将[90]的分类准确率从56.6%提升到60.7%，并且将[91]的准确率从62.7%提升到65.2%，展现了其较强的扩展能力。

基于上述实验，我们可以得出结论：RIDE算法能够产生更强大的具有翻转

	ORIG	RIDE	MI[177]	Max
Pet-37	60.24	**63.49**	58.91	62.12
Aircraft-100	74.61	**78.92**	72.26	76.84
Flower-102	83.53	**86.45**	81.06	85.03
Bird-200	47.61	**50.81**	45.59	49.23

(a) 我们的模型的分类结果

	Pet-37	Aircraft-100	Flower-102	Bird-200
Angelova[181]	54.30	–	80.66	–
Maji[36]	–	48.69	–	–
Murray[85]	56.8	–	84.6	33.3
Paulin[175]	–	–	–	45.2
Pu[182]	–	–	–	44.2
Wang[183]	59.29	–	75.26	–
RIDE	**63.49**	**78.92**	**86.45**	**50.81**

(b) 近期工作的分类结果

表 3.2　我们的算法与近期工作的分类准确率（%）对比。我们在**Aircraft-100**上使用RGB-SIFT特征，在其他三个数据集上使用融合特征（SIFT和LCS）。作为参照，我们也自己实现了MI-SIFT[177]特征。

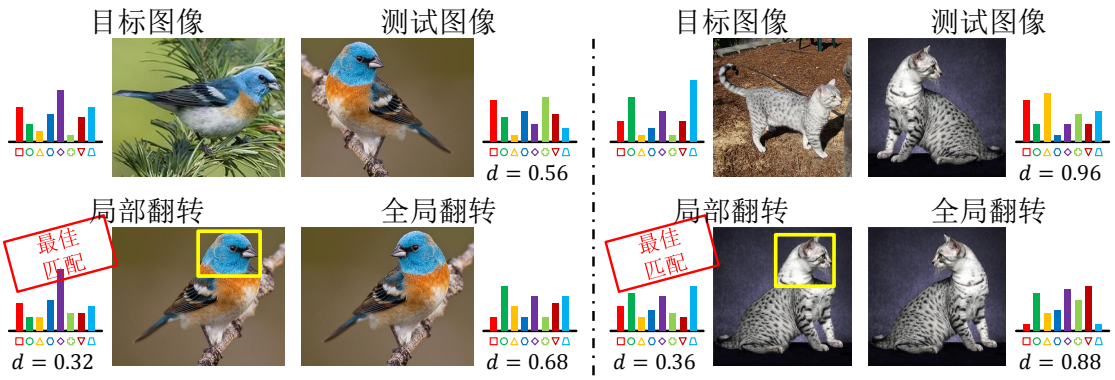


图 3.7　全局翻转与局部翻转的对比。局部翻转能够捕捉小区域内更精确的特征，从而在局部翻转的图像上产生了与原始图像更加接近的表示向量。

不变性的局部特征，并且能够与部件检测算法配合，达到很好的细粒度识别准确率。

	LandUse-21	**Indoor-67**	**SUN-397**
ORIG	93.64	63.17	48.35
RIDE	**94.71**	**64.93**	**50.12**
Junaja[185]	–	63.10	–
Kobayashi[186]	92.8	63.4	46.1
Xie[187]	–	63.48	45.91
Lapin[180]	–	–	49.5

表 3.3　在场景分类数据集上，使用融合特征与RIDE算法产生的分类准确率（%）与近期发表结果的对比。

3.4.4　全局翻转和局部翻转

RIDE算法和数据扩张方法的一个本质区别在于，RIDE算法可以局部翻转图像，而数据扩张方法必须将整张图像一次性翻转。在这里，“局部翻转”的含义是RIDE可以独立地决定每一个描述子是否被翻转，但是数据扩张方法必须使得所有描述子同时被翻转或者不被翻转。图3.7从一种直观的角度比较了两种方法。在展示的两个例子中，我们需要将目标图像与测试图像进行匹配，但是其中的物体的一部分与目标图像的朝向相反。在全局翻转下，我们只有两种选择：或者将整张图像翻转，或者保持原样不动；而采用局部翻转，我们就可以将其中的某些区域（例如鸟类的头部）进行翻转，同时保持其他部分不变。这样，我们就能找到更多的匹配特征对，同时产生更加相似的图像表示。这也是一种直观解释，说明为何RIDE算法能够比数据扩张方法产生更好的分类准确率。

3.4.5　场景识别

为了说明RIDE算法也能够对其他分类任务提供帮助，我们将它应用在场景分类问题上。我们在三个常用的场景分类数据集上进行实验，包括**LandUse-21**数据集（21类航拍场景，2100张图像）[188]、 MIT **Indoor-67**数据集（67种室内场景，15620张图像）[189]、以及**SUN-397**数据集（现今最大的场景数据集之一，397种室内和室外场景，超过10万张图像）[31]。表3.3展示了融合特征（SIFT和LCS）以及RIDE算法产生的分类结果。与之前的情形一样，我们也发现RIDE算法产生了很好的分类效果，超过了近期发表的一些对比方法。

3.4.6　计算复杂度

最后，我们在表3.4中报告视觉词袋模型中每一个模块的计算开销。在应用于SIFT特征上时，RIDE算法只需要计算一个简单的朝向向量**G**，因此它几乎没有

	ORIG	**RIDE**	**AUGM**	**RIDE×2**
特征抽取	2.27小时	2.29小时	2.30小时	2.29小时
建立码本	0.13小时	0.13小时	0.13小时	0.27小时
特征编码	0.78小时	0.78小时	1.56小时	1.28小时
识别任务 （内存开销）	1.21小时 3.71G字节	1.21小时 3.71G字节	2.46小时 7.52G字节	2.42小时 7.51G字节

表 3.4　视觉词袋模型每一个模块的时间和空间开销（单个3.0GHz处理器）。这些数据都在**Bird-200**数据集[34]上产生，使用了SIFT特征和32个GMM分量。

任何额外的时间开销（小于1%）。然而，如果使用数据扩张（**AUGM**）方法，我们必须花费额外的一倍空间存储扩张出的图像数据，同时在在线分类阶段花费双倍的时间和主内存。也就是说，数据扩张方法几乎将使整个系统的时间和空间复杂度加倍。考虑到RIDE算法产生的分类精度并不比数据扩张方法差，低廉的计算开销也成为RIDE算法的另外一个优势。

3.5　本章小结

本章提出了两种算法，即Max-SIFT和RIDE（Reversal Invariant Descriptor Enhancement），其中RIDE是Max-SIFT的扩展。基于对局部描述子在翻转后的变化的观察，我们设计了一种量化标准，以评价一个局部描述子朝向右方的程度。通过强制性地使得局部描述子朝向右方，我们抵消了翻转过程的影响，达到了预期的目的。同时，附录A.2.2节的实验也表明：这种类似归一化的方式对于特征的多样性有明显的损伤，因此应该选择性地加以使用，以防降低识别效率。

图像分类实验表明，RIDE算法显著提升了Max-SIFT和数据扩张方法的分类性能，并且在细粒度物体识别和场景识别任务上都能很好地工作。与传统数据扩张方法相比，Max-SIFT和RIDE算法还具有时间和空间复杂度较低的特点。我们希望这种算法能够得到更多的关注和更广泛的应用。

第 4 章　局部特征的强化编码

4.1　研究动机

图像分类的一个重要步骤是对局部特征进行编码（见第2.2.3节）。在传统的视觉词袋模型[10]中，编码过程的作用不仅体现在将所有特征压缩为同样长度的向量，还在于将这些特征量化为特征空间中的规范化表示，从而使得特征组合算法产生的图像表示具有更强的鲁棒性。典型的特征编码有硬量化（Hard Quantization，HQ）、软量化（Soft Quantization、SQ）、局部限制的线性编码（Locality-constrained Linear Coding，LLC）[9]、超向量编码（Super Vector encoding，SV）[81]、*Fisher*向量（Fisher Vector，FV）[7]编码，等等。这些编码方式大多基于事先训练的码本，通过捕捉局部特征之间的关系，将每个描述子编码为一个高维的向量。

虽然上述方法大都取得了很好的分类效果，然而总体来说，基于局部特征编码的算法仍然存在较大的缺陷。这些缺陷大多源于众所周知的、介于低级描述子与高级视觉概念之间的语义鸿沟（semantic gap）[16]，一个显见的例子就是SIFT特征同时受到同义性（synonymy）和多义性（*polysemy*）[17]的影响。局部描述子的描述力不足限制了图像表示的准确性，从而导致分类准确率的下降。为了解决这些问题，有如下几种常见的手段：

- **使用多种不同的局部特征。** 由于单一特征通常不能捕捉图像中丰富的语义信息，抽取多种互补性的特征就成为一种常见的做法。多种特征融合后的表示通常更加鲁棒，更具有区分性。例如，通过简单地将纹理和形状的特征向量拼接在一起，[18]就能获得比采用单一特征更好的分类精度。
- **使用中级结构连接低层描述子和高层概念。** 在词袋模型中，图像通常表示为一个视觉单词的集合，然而语义鸿沟的存在[16]限制了这种方法的表示能力。因此，许多研究人员建议使用一些中级的结构来进行过渡。典型的例子如宏描述子（macro descriptors）[190]和视觉短语（visual phrases）[17][21]等。
- **图像平面上的加权综合。** 在一张图像中，并非所有的区域都是同等重要的，例如背景区域通常会出现噪声，因而降低模型训练的稳定性。因此，检测图像中的兴趣区域（Regions-of-Interest，RoI）就显得十分重要。典型的例子如[18]和[86]，它们针对图像中较小的兴趣物体往往非常有效。

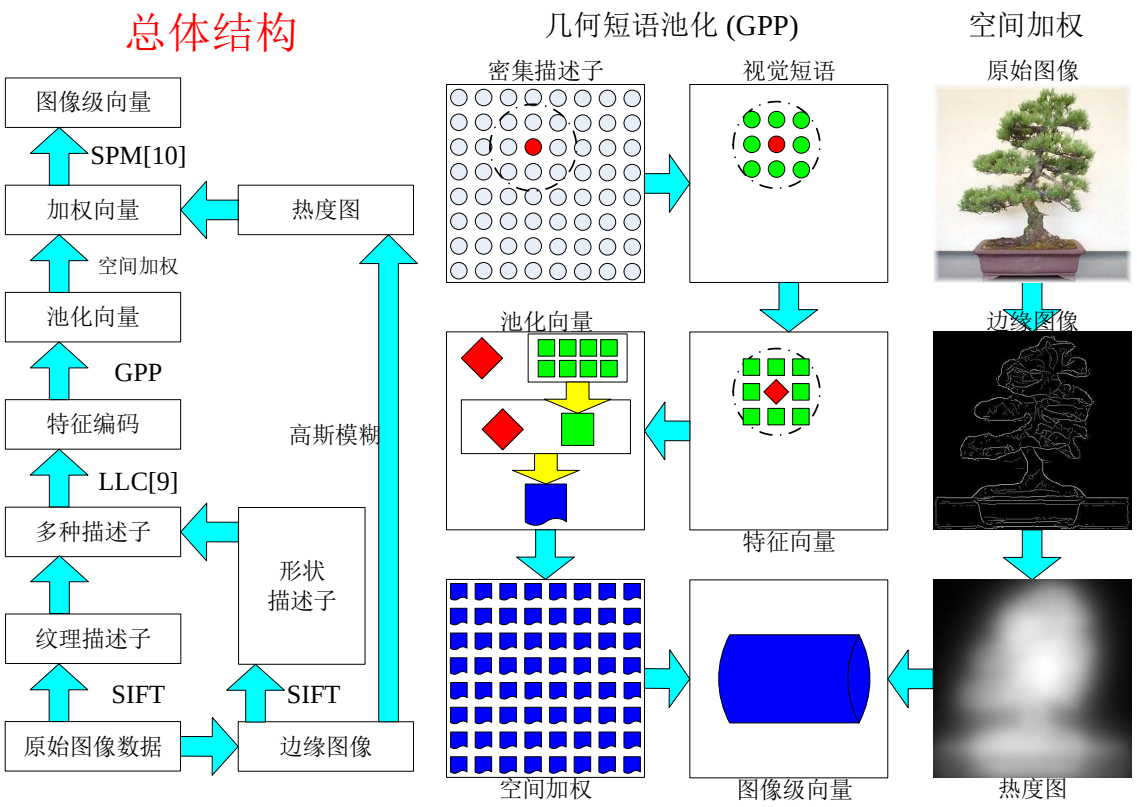


图 4.1　我们提出的改进词袋模型以及若干新模块。左边：扩展的词袋模型；右边：几何短语池化（Geometric Phrase Pooling，GPP）和基于边缘的空间加权算法。

本章主要提出若干新算法，包括一种与SIFT特征能够直接融合的**边缘SIFT**特征、**几何短语池化**（Geometric Phrase Pooling，GPP）算法、以及**基于边缘的空间加权**算法。此外，我们将这些算法组合成一个完整的系统，使得它们能够互相配合，增加图像表示的描述力。我们设计的分类框架如图4.1所示。我们将算法在多个不同的图像数据集上进行测试（第4.5节），达到了很好的分类效果。

与本章相关的出版物为[191][71]。

4.2　提取互补的局部特征

在这一节里，我们提出一种新的抽取互补特征的方法。我们首先提出一种称为**边缘SIFT**（Edge-SIFT）的特征，然后将同一张图像上提取的SIFT和边缘SIFT特征合并为一个集合。它们虽然来源不同，但是对应维却具有相同的物理意义（梯度直方图），因此能够直接结合起来，增强图像的表示效果。最后，我们讨论这种方法的局限性。

4.2.1　SIFT和Edge-SIFT特征

对于一张$W \times H$的图像$\mathbf{I}$，我们在其上抽取密集SIFT特征[4]。将这些SIFT特征表示为一个集合$\mathcal{D}^{\mathrm{S}}$：

$$\mathcal{D}^{\mathrm{S}} = \left\{\left(\mathbf{d}_1^{\mathrm{S}}, \mathbf{l}_1^{\mathrm{S}}\right), \left(\mathbf{d}_2^{\mathrm{S}}, \mathbf{l}_2^{\mathrm{S}}\right), \ldots, \left(\mathbf{d}_{M^{\mathrm{S}}}^{\mathrm{S}}, \mathbf{l}_{M^{\mathrm{S}}}^{\mathrm{S}}\right)\right\} \tag{4-1}$$

其中，上标S表示“SIFT”，M^{S}是图像上SIFT特征的数量。

研究表明，SIFT特征能够有效地表达纹理信息，但是对于形状特征的表示较弱。为了克服这样的缺点，我们引入一种新的用于形状描述的特征以辅助原始SIFT特征。遵循[18]我们在图像$\mathbf{I}$上计算边缘响应，得到另一个$W \times H$的灰度图像$\mathbf{I}^{\mathrm{E}}$：

$$\mathbf{I}^{\mathrm{E}} = \left(e_{ij}\right)_{W \times H} \tag{4-2}$$

这里e_{ij}是一个[0, 1]范围内的浮点数，表示像素(i, j)在边缘上的可能性，或者边缘响应强度。我们称$\mathbf{I}^{\mathrm{E}}$为与原图$\mathbf{I}$对应的边缘图像（edgemap）。我们采用Compass算子（Compass Operator）[149]进行边缘提取。图4.2、图4.4、图4.5和图4.6中展示了一些边缘提取的例子。在边缘图上，物体的纹理特征大部分被筛去，形状特征变得更加明显。因此，我们可以在边缘图上提取SIFT特征用于描述原图的形状信息。我们称边缘图上提取的SIFT特征为边缘SIFT（Edge-SIFT），表示为集合$\mathcal{D}^{\mathrm{E}}$：

$$\mathcal{D}^{\mathrm{E}} = \left\{\left(\mathbf{d}_1^{\mathrm{E}}, \mathbf{l}_1^{\mathrm{E}}\right), \left(\mathbf{d}_2^{\mathrm{E}}, \mathbf{l}_2^{\mathrm{E}}\right), \ldots, \left(\mathbf{d}_{M^{\mathrm{E}}}^{\mathrm{E}}, \mathbf{l}_{M^{\mathrm{E}}}^{\mathrm{E}}\right)\right\} \tag{4-3}$$

类似地，上标E表示“边缘SIFT（Edge-SIFT）”，M^{E}表示边缘SIFT特征的数量。注意M^{E}可能不同于M^{S}，因为两者的采样方式（如空间跨度）可能存在差别。

SIFT和边缘SIFT特征都是基于梯度的直方图。虽然来源不同，对应维度却具有相同的物理含义（相应位置相应方向上的梯度强度），因此我们可以在后续操作中将它们直接结合在一起。

4.2.2　融合两种特征

我们在图像上直接将这两种特征融合起来：

$$\mathcal{D} = \mathcal{D}^{\mathrm{S}} \cup \mathcal{D}^{\mathrm{E}} \tag{4-4}$$

这里，$\mathcal{D}$是融合后的特征集合，其中的特征数量M满足$M = M^{\mathrm{S}} + M^{\mathrm{E}}$。

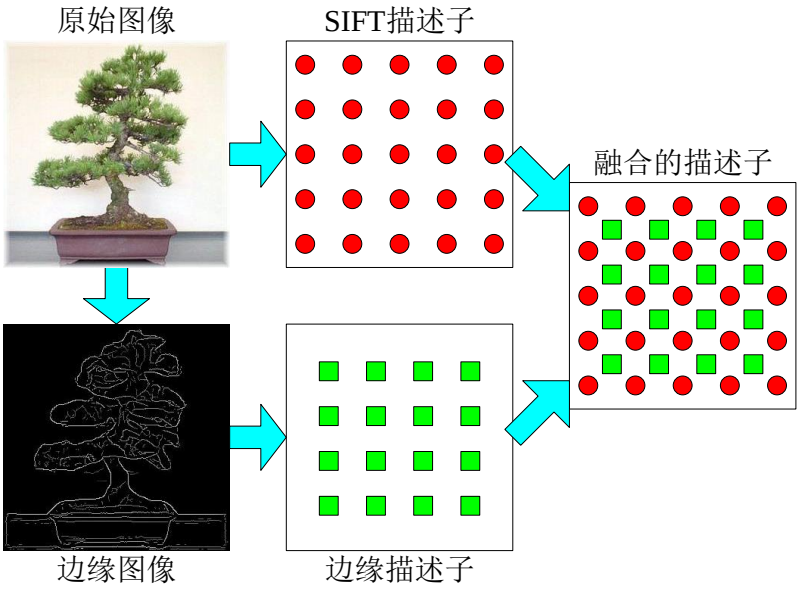


图 4.2　在图像上融合两种不同的特征。

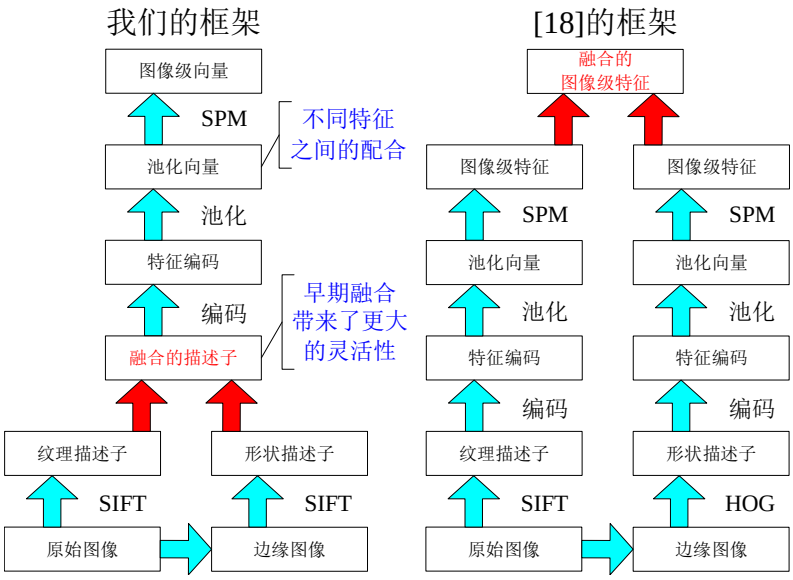


图 4.3　我们的模型和[18]中提出的模型的区别。红色的箭头和文字高亮表示了每一个模型中融合的步骤。

我们将融合过程表示为图4.2。这里我们需要强调：融合过程保留了所有描述子的向量和位置信息，在第4.3.5节，我们将论述这种策略所带来的好处，特别是能够建立同时包含纹理和形状信息的几何视觉短语。

除了原始图像和边缘图像外，我们还可以计算其他的图像（如等高图像、显著性图像等），并且在其上提取SIFT特征。只要我们能够确保提取的特征的对应维度具有相同的物理意义，就可以将这些特征融合起来，形成一个更大的特征集合。但在本文接下来的部分，我们只考虑SIFT特征和边缘SIFT特征的融合。

我们简要地对比我们的工作和[18]中提出的模型。在[18]里，作者同样建议在

特征#1	特征#2	只用#1	只用#2	融合#1和#2
SIFT(7,7)	SIFT(6,12)	74.41%	72.69%	75.14%
SIFT(7,7)	Edge(6,12)	74.41%	73.08%	78.75%
SIFT(7,7)	SIFT(7,12)	74.41%	73.77%	75.75%
SIFT(7,7)	Edge(7,12)	74.41%	72.89%	**78.94**%
SIFT(7,7)	SIFT(8,12)	74.41%	73.60%	75.32%
SIFT(7,7)	Edge(8,12)	74.41%	72.93%	78.92%

表 4.1　使用不同特征，在**Caltech101**数据集上的分类准确率。小括号内的数字分别表示抽取特征时的空间跨度和窗口大小。

原始图像和边缘图像中同时提取相同类型的特征（如SIFT），但是这两种特征在整个视觉词袋模型计算过程中（如聚类、编码等模块里），都被独立地看待和处理，直到分类之前才将两个向量拼接在一起。这种较高层次的融合将限制模型的灵活度，从而限制中级结构（如视觉短语）的描述能力，进而影响模型的分类效果。相反地，我们的模型在很早的环节（提取局部特征之后）就进行了融合操作，使得后续算法的空间增大。这两种模型的区别表示在图4.3中，我们也将在第4.3.5节详细叙述提前融合模型的优势。

4.2.3　实验和讨论

我们在**Caltech101**数据集[28]上测试我们的算法。基础的设定参见第4.5.1节。我们使用两种局部特征（SIFT和边缘SIFT）来描述图像信息。我们在数据集中的每一类随机选取30张训练图像，并且随机进行10次训练和测试数据的划分，然后报告平均的分类精度（按类平均）。

表4.1展示了不同特征融合的分类结果。可以发现，最佳的分类准确率出现在两种不同的特征（SIFT和边缘SIFT）融合的结果中。虽然用同一种特征的两个不同集合进行融合也能够提高分类精度，但是显然使用不同特征达到的效果更好，因为我们可以捕捉到更多的互补信息（纹理和形状）。

为了提供进一步的分析结果，我们回到使用单一特征的分类系统。我们从表4.1中选择融合之后产生最优分类结果的参数，即SIFT和边缘SIFT特征都采用7个像素作为空间跨度、SIFT特征使用7×7的窗口大小、边缘SIFT特征采用12×12的窗口大小。我们分别测试独立采用两种特征的分类系统，按类对比它们的分类精度，找到其中SIFT和边缘SIFT分类精度差别最大的几类，显示在表4.2和图4.4 中。

很明显，不同类型的物体可以被不同类别的特征更好地描述出来。直观上来

类名	SIFT特征	Edge特征	差别
wild cat	57.50%	27.50%	30.00%
water lily	65.71%	38.57%	27.14%
crocodile	33.00%	13.00%	20.00%
ferry	89.46%	70.81%	18.65%
hedgehog	82.50%	65.83%	16.67%
anchor	44.17%	73.33%	29.17%
butterfly	50.16%	72.62%	22.46%
wrench	57.78%	77.78%	20.00%
pyramid	71.48%	87.04%	15.56%
saxophone	75.00%	90.00%	15.00%

表 4.2　**Caltech101**数据集上采用两种不同类型特征的分类结果对比。表格的上下两部分分别是用SIFT和用边缘SIFT分类效果更好的类别。

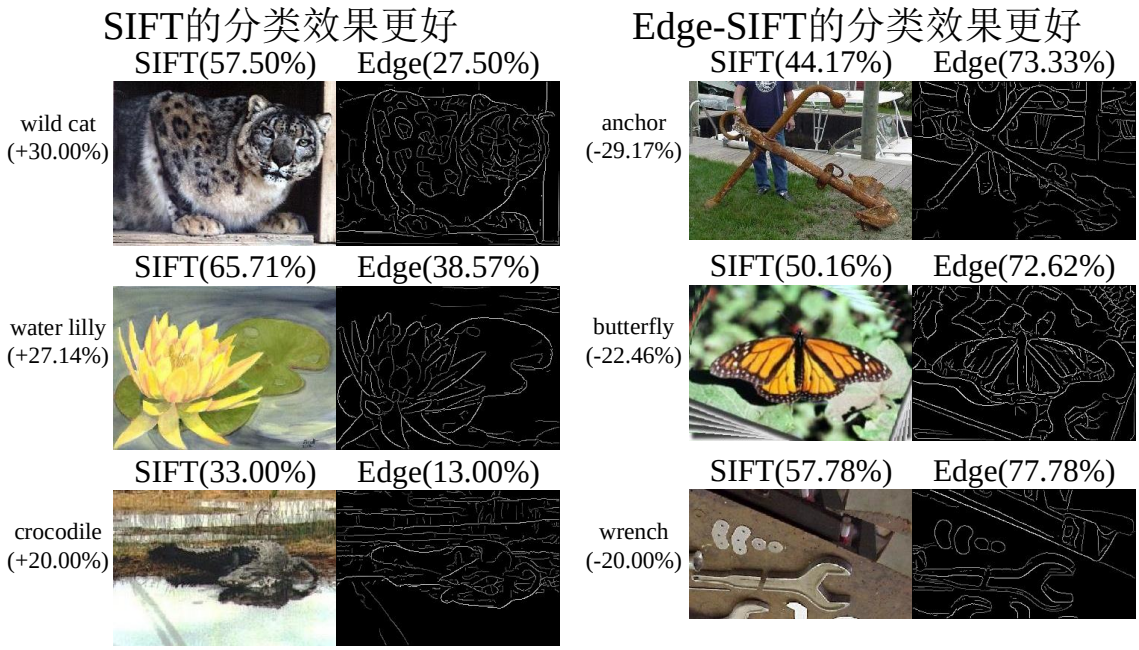


图 4.4　两种特征的来源对比。在每一对图像的上方，我们分别陈列了两种特征在相应类别上的分类准确率。

说，对于那些形变较小的类（如人工制造的工具），忽略纹理特征而采用形状特征能够得到更好的分类结果；相反，对于那些纹理特征较为丰富的类（如动植物），直接在原图上提取SIFT特征可能会达到更好的效果。

由于SIFT特征和边缘SIFT特征具有以上的互补性，将它们同时保留下来就成为了很自然的想法。图4.5展示了同时保留两种特征的好处。在图示的4个类别中，*panda*和*football*的图像都包含一些黑白相间的纯色图像块，然而*panda*图像中

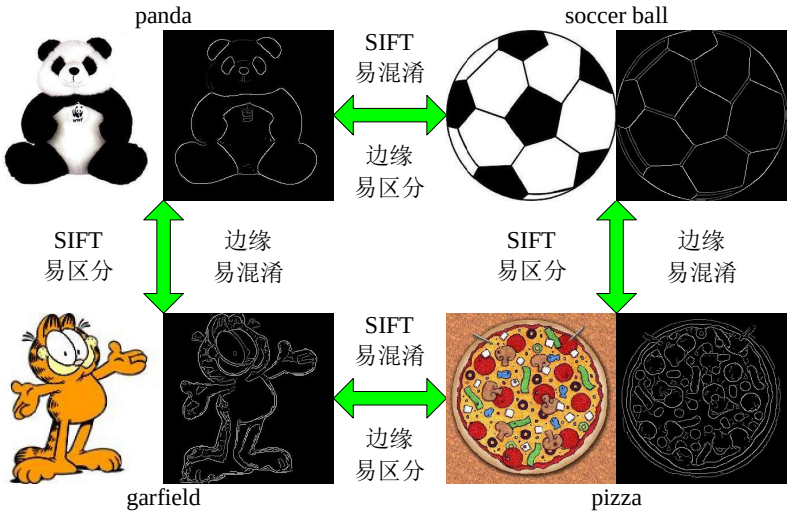


图 4.5　融合后的特征有助于区分那些容易混淆的类别。

特征#1	特征#2	只用#1	只用#2	融合
SIFT(8,16)	Edge(10,16)	69.51%	27.23%	65.48%
SIFT(8,16)	Edge(12,16)	69.51%	28.66%	64.90%
SIFT(8,16)	Edge(16,16)	69.51%	27.43%	66.72%
SIFT(12,16)	Edge(10,16)	69.52%	27.23%	63.34%
SIFT(12,16)	Edge(12,16)	69.52%	28.66%	63.55%
SIFT(12,16)	Edge(16,16)	69.52%	27.43%	65.56%

表 4.3　利用单一特征和融合特征，在Oxford **Flower-17**数据集上的分类结果。括号内的数字仍然表示密集SIFT采样的空间跨度和窗口大小。

出现的线条更有可能是曲线，而*football*图像则可能包含更多的直线段。因此，边缘SIFT对于区分这两个类更有帮助。相似的结论在其他几个类的对比中也能够观察到。因此，对于图示的这四个类别，使用单一的特征不容易将它们区分开来，而使用两种特征则能够显著提高分类的精度。

4.2.4　局限性

虽然我们提出的融合特征方法在**Caltech101**数据集上取得了很好的分类效果，但这并不意味着它在其他的情况下也能很好地工作。为此，我们在Oxford **Flower-17**数据集[32]上测试特征融合方法。测试的方式类似于**Caltech101**数据集，结果如表4.3所示。我们观察到，此时最佳的分类准确率来源于采用单一的SIFT特征：无论单独使用边缘SIFT或者融合两种特征，都降低了分类精度。

为了解释这一现象，我们仿照先前的分析方式，对**Flower-17**的每一类单独分析其准确率。结果如表4.4所示。容易看出，SIFT特征在所有的17个类上，都取得

Category	SIFT	Edge	Category	SIFT	Edge
001	**66.27**%	16.13%	002	**59.07**%	22.53%
003	**58.80**%	22.40%	004	**68.80**%	10.93%
005	**43.73**%	18.53%	006	**68.40**%	48.13%
007	**81.07**%	23.20%	008	**26.53**%	7.47%
009	**80.40**%	44.40%	010	**95.33**%	50.53%
011	**93.73**%	22.27%	012	**44.53**%	22.53%
013	**83.60**%	52.80%	014	**61.07**%	15.73%
015	**87.33**%	22.53%	016	**84.53**%	43.60%
017	**78.67**%	43.47%			

表 4.4　在Oxford **Flower-17**数据集上采用不同特征的分类结果对比。采用单独的SIFT特征在所有情况下都取得了更好的结果。

了比边缘SIFT特征更高的准确率。这一现象与我们在**Caltech101**数据集上的实验结果不同：SIFT和边缘SIFT各自在某些类上工作得更好。显然，这样的实验现象与**Flower-17**数据集的特点有关（如图4.6所示）。由于这个数据集里的物体种类比较单一（所有图像都是花），在区分不同类别的花时，纹理特征比形状特征重要得多。因此，如果将纹理特征筛除而使用边缘SIFT特征，我们就将忽略至关重要的纹理信息，从而无法取得良好的分类结果。即使将两种特征融合在一起，描述力显著不足的边缘SIFT特征也将产生大量不稳定特征，从而使得我们更难训练出一个有效的分类模型。

当然我们必须承认，所有特征（除了随机噪声）都能够对图像分类产生一定的帮助。然而，在所有的数据集里，训练图片的数量都是有限的。如果抽取的特征并非足够有效，那么增加的维度就会给训练过程带来过拟合的风险。上述实验结果也启发我们，当考虑将两种特征融合在一起的时候，可以先独立地对它们进行测试，如果测试的结果表明某种特征在所有的类上都产生了比另一种特征高得多的分类准确率，那么融合往往就是不必要的。

作为上述情况的推广，我们关注近年来非常流行的细粒度分类问题。细粒度分类数据集（如Caltech-UCSD **Birds-200**-2011[34]）通常包含大量形状非常相近的物体。此时，纹理特征产生的作用通常比形状特征要大得多[192][90][91]。因此，抽取形状特征，无论是单独使用还是与纹理特征融合，都会对分类精度产生影响。在这些数据集上，融合特征之前都应当对不同的特征进行独立测试，以确保它们能够具有互补性。

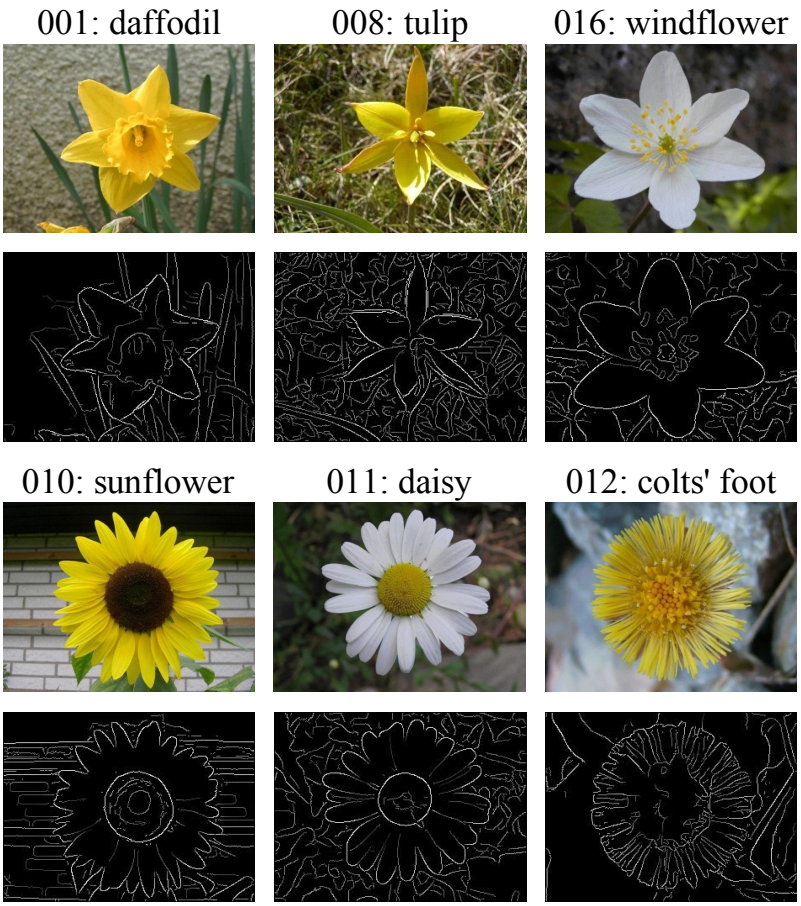


图 4.6　在**Flower-17**数据集上，SIFT特征总是比边缘SIFT特征工作得更好，因为后者忽略了对花朵分类至关重要的纹理信息。同一组的上下两行展示了同一个样例的原始图像和边缘图像。

4.3　几何短语池化

在前面的章节中，我们描述了一种简单的抽取图像中互补特征的算法。我们同时注意到，视觉词袋模型对于每个描述子的处理是独立的，这不利于建立一些中层结构，来辅助图像表示。本节将提出一种**几何短语池化**（Geometric Phrase Pooling，GPP）算法。它基于这样的观察：位于一个局部范围内的特征之间可能存在一定的相关性，如果能够为它们建立一个紧凑的特征编码，就能够捕捉更多的信息，对图像分类产生帮助。我们建立的几何视觉短语结构和相应的编码算法，可以为图像建立起一种中层表示结构，并用于后续的处理过程。图4.1展示了GPP算法的大致流程。

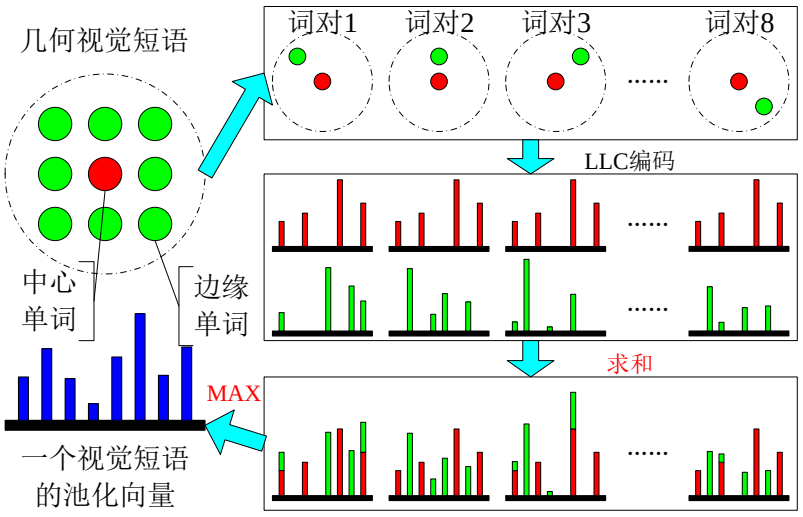


图 4.7　几何短语池化（Geometric Phrase Pooling，GPP）算法的直观展示。我们将中心词语和边缘单词看作一个组合，将它们的特征向量相加，随后在所有的求和向量上按维求最大值，即得短语池化向量。

4.3.1　GPP算法

我们从(4-4)式入手，将其重写为更简单的形式：

$$\mathcal{D} = \{(\mathbf{d}_1, \mathbf{l}_1), (\mathbf{d}_2, \mathbf{l}_2), \ldots, (\mathbf{d}_M, \mathbf{l}_M)\} \tag{4-5}$$

我们首先给出一个全新的**几何视觉短语**（Geometric Visual Phrases，GVP）的定义。遵循[17]，我们将几何视觉短语定义为一些在图像空间上位置接近的视觉单词的组合。由于图像中的视觉词语并不像文本中的单词一样具有强词序信息，我们简单地忽略视觉短语中单词的顺序，将其当成一个无序的单词组合。为此我们确定一个整数K，$K \ll M$，并且寻找每个单词在图像平面上的K个最近邻（以欧氏距离评价），依此为每个描述子构建一个局部的特征组（视觉单词）：

$$\mathcal{G}_m = \{(\mathbf{d}_{m,0}, \mathbf{l}_{m,0}), \ldots, (\mathbf{d}_{m,K}, \mathbf{l}_{m,K})\} \tag{4-6}$$

这样，$\mathcal{G}_m$就称为第m个几何视觉短语。$\mathcal{G}_m$的中心单词（central word）定义为$(\mathbf{d}_{m,0}, \mathbf{l}_{m,0})$，即$(\mathbf{d}_m, \mathbf{l}_m)$本身，而其他的$K$个单词则称为边缘单词（side words）。短语$\mathcal{G}_m$的位置定义为$\mathbf{l}_{m,0}$，它的大小定义为K。

假设我们已经训练了一个具有B个单词的码本$\mathcal{B}$。对于一个视觉短语$\mathcal{G}_m$，我们对它的每一个单词计算LLC编码[9]。LLC是一种稀疏编码，在给定近邻参数r（通常$r \ll B$）的情况下，编码所得的B维向量里至多有r个非零元素。这样，我们就在$\mathcal{G}_m$里得到了$K+1$个稀疏向量，与$K+1$个视觉单词一一对应。将$\mathcal{G}_m$里第k个单词的编码向量记为$\mathbf{w}_{m,k}$。

在上述基础上，几何短语池化算法很容易实现。我们计算一个B维向量$\mathbf{p}_m$作为$\mathcal{G}_m$的编码：

$$\mathbf{p}_m = \max_{1\le k\le K}\{\mathbf{w}_{m,0}+\mathbf{w}_{m,k}\} \tag{4-7}$$

$$= \mathbf{w}_{m,0}+\max_{1\le k\le K}\mathbf{w}_{m,k} \tag{4-8}$$

其中，符号$\max_k$表示K个向量按维取最大值。方程(4-7)是GPP的核心公式，而方程(4-8)是它的一种等价的，更容易实现的形式。我们将方程(4-7)的工作机理表示为图4.7。

值得注意的是，GPP是一个介于局部特征编码和综合之间的模块。在GPP处理之后，我们仍然需要对所有短语进行池化操作（如最大池化），以得到图像的全局表示向量$\mathbf{F}$：

$$\mathbf{F}^{\mathrm{GPP}} = \max_{1\le m\le M}\mathbf{p}_m \tag{4-9}$$

4.3.2　GPP的深入解释

GPP的公式(4-7)很容易实现，但是其中蕴含的道理却并不直观。为了解释，我们回到(4-7)式和(4-9)式，并且利用简单的推导将它们改写成如下形式：

$$\mathbf{F}^{\mathrm{GPP}} = \max_{1\le m\le M}\mathbf{p}_m \tag{4-10}$$

$$= \max_{1\le m\le M}\left\{\max_{1\le k\le K}\{\mathbf{w}_{m,0}+\mathbf{w}_{m,k}\}\right\} \tag{4-11}$$

$$= \max_{1\le m_1,m_2\le M,m_1\diamond m_2}\{\mathbf{w}_{m_1}+\mathbf{w}_{m_2}\} \tag{4-12}$$

这里，$m_1\diamond m_2$意味着视觉单词$\mathbf{w}_{m_1}$和$\mathbf{w}_{m_2}$是一个“临近单词对”：当我们将其中之一选定为中心单词时，另外一个就会成为相应短语里的边缘单词。这样，全局最大池化所得的向量$\mathbf{F}^{\mathrm{GPP}}$就可以解释为所有临近单词对的编码之和，按维取最大值。我们很自然地将$\mathbf{w}_{m_1}+\mathbf{w}_{m_2}$定义为近邻单词对$(m_1,m_2)$对于$\mathbf{F}^{\mathrm{GPP}}$的贡献。

如果不使用GPP而只进行最大池化，我们也可以将其最终的表示向量$\mathbf{F}^{\mathrm{MAX}}$写成类似的形式（虽然包含一定的冗余性）：

$$\mathbf{F}^{\mathrm{MAX}} = \max_{1\le m\le M}\mathbf{w}_m \tag{4-13}$$

$$= \max_{1\le m\le M}\left\{\max_{1\le k\le K}\{\max\{\mathbf{w}_{m,0},\mathbf{w}_{m,k}\}\}\right\} \tag{4-14}$$

$$= \max_{1\le m_1,m_2\le M,m_1\diamond m_2}\{\max\{\mathbf{w}_{m_1},\mathbf{w}_{m_2}\}\} \tag{4-15}$$

自然地，$\max\{\mathbf{w}_{m_1},\mathbf{w}_{m_2}\}$就成为近邻单词对$(m_1,m_2)$对于$\mathbf{F}^{\mathrm{MAX}}$的贡献。显然，(4-12)式和(4-15)式之间的唯一区别，就是近邻单词对在$\mathbf{F}^{\mathrm{GPP}}$和$\mathbf{F}^{\mathrm{MAX}}$中的贡献是不同

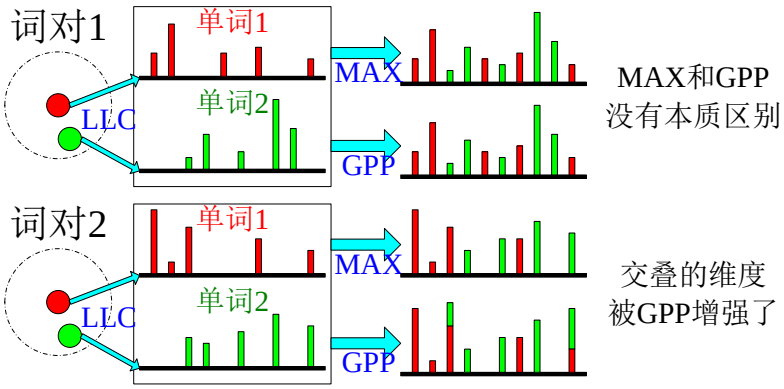


图 4.8　直观地解释GPP为特征编码带来的不同。我们展现了非重叠和重叠的近邻单词对的不同情况，GPP增强了重叠维度的响应值。

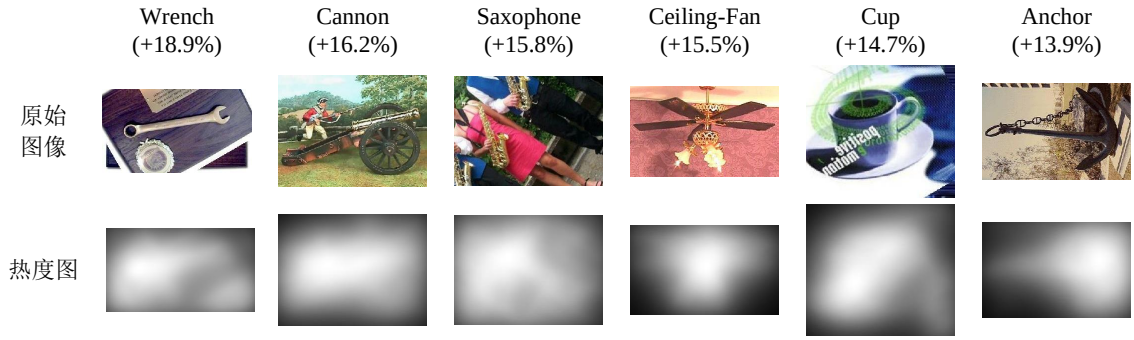


图 4.9　根据近邻视觉单词对中有用部分的比例，在图像平面上产生的热度图。括号内的数字表示相应类别在使用GPP进行编码后，比仅使用LLC编码的精度增长。

的。显式地，对于单词对(m_1, m_2)，以及它对于两者的贡献：$\mathbf{w}_{m_1} + \mathbf{w}_{m_2}$（对$\mathbf{F}^{\text{GPP}}$）和 $\max\{\mathbf{w}_{m_1}, \mathbf{w}_{m_2}\}$（对$\mathbf{F}^{\text{MAX}}$）。如果$\mathbf{w}_1$和$\mathbf{w}_2$并不包含任何重叠的非零维度，则显然有$\max\{\mathbf{w}_1, \mathbf{w}_2\} = \mathbf{w}_1 + \mathbf{w}_2$。此时，单词对$(m_1, m_2)$对于两者的贡献是完全相同的。如果$\mathbf{w}_1$和$\mathbf{w}_2$中包含重叠的非零维度，那么这个维度的响应在GPP下会被增强，造成两种编码方式的结果不同。图4.8直观地表达了我们的解释。

在接下来的部分，我们分析非零维度的重叠对分类问题带来的好处。注意到我们采用的LLC是一种局部敏感的编码算法，如果两个特征向量出现了维度重叠，这说明它们在特征空间中的位置也是非常相近的，即这两个特征非常相似。也就是说，GPP算法选择性地增强了那些在图像平面和特征空间上都近邻的特征对。

我们知道，在自然图像中，相关的局部特征更有可能具有相似的表观（在特征空间中相似）。因此对于包含一个中心单词和K个边缘单词的几何视觉短语，与中心单词相似的边缘单词更有可能对短语的编码产生重要的作用。我们利用如下的实验来展示我们的推断。对于每个视觉短语，我们计算其中与中心单词相关（有非零维度）的边缘单词的比例，并且在图像平面上将上述比例表示为热度图（heatmap）。图4.9展示了**Caltech101**数据集上的若干图像以及它们对应的热度图。

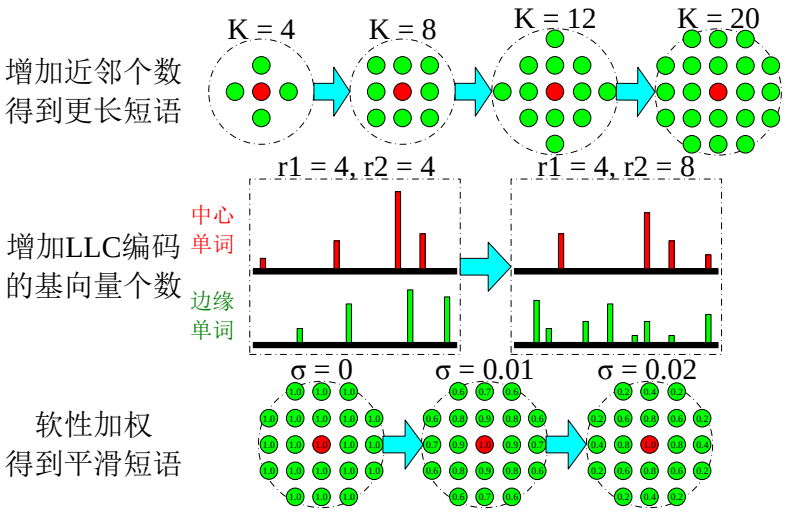


图 4.10　三种提升GPP效果的方法。上部：增加K以获得更大的视觉短语；中部：增加LLC编码基的数目，以增加相关的单词对的数量；下部：降低较远的边缘单词的权值。

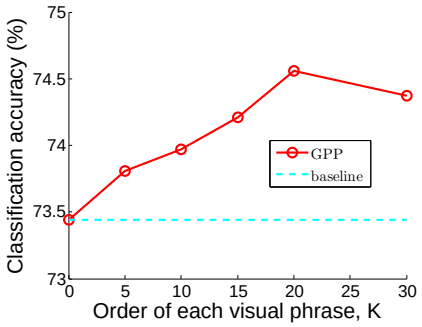


图 4.11　视觉短语的大小对于分类结果的影响。虚线表示采用基准算法（LLC）的分类结果73.44%。

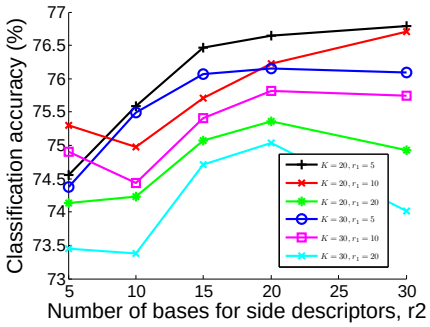


图 4.12　LLC编码的基的个数也会影响分类效果。我们将不同的K和r_1（中心词语的基）的效果用不同的折线表示。

可以发现，被显著增强的那些区域大多位于图像的感兴趣区域（即包含一定量的语义信息）。通过增强这些位置的特征响应，GPP能够比原先的LLC算法更加精确地捕捉具有语义的视觉线索。

因此，GPP可以被视为一种有效的中层视觉概念，它具有直观性强和实现简单的优点。

4.3.3　增强GPP的效果

本节将提供三个有效的方法以提升GPP的效果，这些方法被总结在图4.10中。本节的所有实验仍然在**Caltech101**数据集上进行。

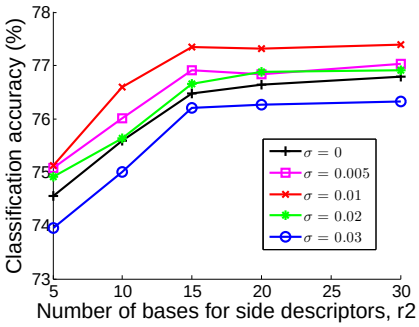


图 4.13　平滑参数σ_w的作用（在图中简记为σ）。我们提供$\sigma_w = 0$（不做平滑）的结果作为对比。

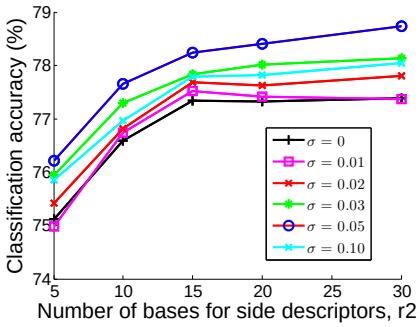


图 4.14　平滑参数σ_e的作用（在图中简记为σ）。我们提供$\sigma_e = 0$（不做边缘平滑）的结果作为对比。

4.3.3.1　更大的视觉短语

在第4.3.2节中我们观察到：只有那些同时在图像平面和特征空间中近邻的单词对才能真正对GPP产生贡献。为了寻找更多这样的单词对，我们可以增大K值，以相应增加每个视觉短语的中心单词-边缘单词对的数量。然而，如果视觉短语变得太长，许多与中心单词距离较远的边缘单词也会被添加进来，引入不稳定的噪声。在实际操作中，我们在5到30范围内选择K，分类结果如图4.11所示。在接下来的实验中，我们固定$K = 20$。

4.3.3.2　更多的编码基

LLC是一种稀疏编码，其中的编码基个数r一般远小于码本的大小B。在这种情况下，很少有特征对能够具有重叠的非零维度。为了让对GPP产生贡献的单词对更多，我们增加编码基的数量。由于太多的编码基会导致编码的不稳定，因此我们需要在描述力和稳定性之间做出平衡。

中心单词和边缘单词在视觉短语中的地位是不同的：中心单词的作用更加重要，因此对于稳定性的需求也更大。因此，我们提出对于中心单词和边缘单词采用不同数量的编码基。将中心单词和边缘单词的编码基数量分别记为r_1和r_2，一般有$r_1 < r_2$。不同的r_1和r_2组合的分类结果如图4.12所示。遵循实验结果，我们在后续的实验中固定$r_1 = 5$和$r_2 = 30$。

4.3.3.3　更加平滑的加权方式

根据自然图像的性质，两个视觉单词相距较远时，它们的视觉关系就相应较弱。因此，我们利用一种类似指数递减的方式，对边缘单词进行加权，以弱化较

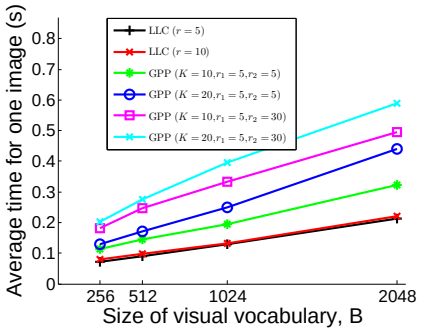


图 4.15　每张图像编码的平均时间。测试在4核2.0GHz的CPU上进行。

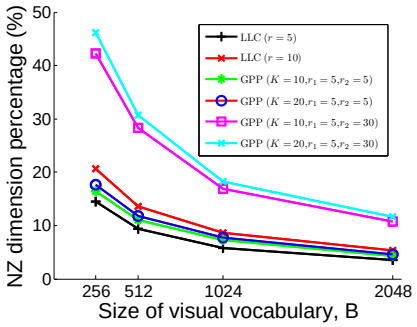


图 4.16　池化后的特征的稀疏性（平均非零元素比例）对比。

远的边缘单词的影响。在实际操作中，我们对于一个边缘单词$(\mathbf{d}_{m,k}, \mathbf{l}_{m,k})$加权$s_k$：

$$s_k = \exp\{-\sigma_w \times \|\mathbf{l}_0 - \mathbf{l}_k\|_2\} \tag{4-16}$$

其中，σ_w是平滑参数，而$\|\cdot\|_2$表示欧氏距离。这样，原始的GPP算式(4-8)就被修改为一个更加平滑的版本：

$$\mathbf{p}_m = \mathbf{w}_{m,0} + \max_{1 \le k \le K} s_k \times \mathbf{w}_{m,k} \tag{4-17}$$

为了选择一个合适的平滑参数，我们测试不同的σ_w的分类效果，分类结果如图4.13所示。实验中得到的最好结果（$\sigma_w = 0.01$）将被应用于后续实验中。

本节提供的三种增强GPP效果的方法（加长短语、增加编码基、平滑加权）可以互相配合。从图4.11–图4.13的实验结果看， GPP对于每个独立的参数并不是非常敏感，因为其中最好结果和较好结果之间的差别并不是非常大（大约1%）。通过最佳的参数组合（$K = 20$、$r_1 = 5$、$r_2 = 30$、$\sigma_w = 0.01$），我们在**Caltech101**数据集上达到了77.39%的分类精度，显著地超过了LLC算法的73.44%。虽然这些参数大多是在**Caltech101**数据集上测试的，但是它们在很多其他图像分类数据集上也工作得很好。

4.3.4　时间复杂度和稀疏性

我们测试GPP算法的时间复杂度，以及它产生的特征向量的稀疏性。为了对比，我们在不同大小的码本上同时计算LLC编码和GPP编码。注意到LLC编码和GPP编码产生的特征向量长度相同，因此只需要比较两者的非零元素个数。

GPP算法的时间开销和稀疏性如图4.15和图4.16所示。由于编码算法的简单性（只需求和与求最大值），GPP在LLC编码基础上需要的额外时间开销几乎可以忽略不计。平均计算，当基向量个数为5和30时，LLC算法在每张图像上分别需

	后期融合[18]	早期融合（我们的算法）
LLC[9]	79.12%	78.94%
GPP (Ours)	79.48%	**81.36**%

表 4.5　不同的融合与编码算法在**Caltech101**数据集上的分类结果。

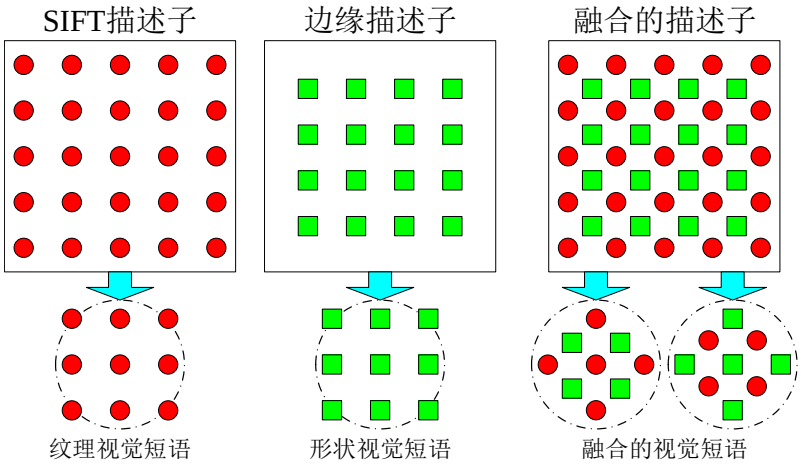


图 4.17　将GPP算法应用在不同来源的视觉单词上，使我们能够编码更具描述性的局部特征组。

要0.2秒和0.4秒。因此，GPP（$r_1 = 5$，$r_2 = 30$）所需要的总时间为0.6秒，大约是原始LLC算法的3倍，远小于某些复杂的编码算法，如几何p范数池化（Geometric ℓ_p-norm Pooling，GLP）[86]。另一方面，图4.16表明GPP产生的特征比LLC编码要密集得多，尤其是在几何短语较长，基向量个数较多的情况下。近年来，有学者指出，稀疏的向量在分类问题上工作得更好[78]。我们的实验提供了这一论述的另一方面观点：密集的特征并不一定会产生较差的结果，我们只需要一种更合理的方式来组织这些特征。 GPP在不改变特征长度的情况下，使特征更加密集，从而编码了更多对分类有益的信息。其他一些密集特征编码算法[7][81]也佐证了我们的观点。

4.3.5　早期融合与后期融合

最后，我们为第4.2.2节关于早期融合与后期融合的讨论提供进一步的说明。在**Caltech101**数据集上，我们使用LLC和GPP对特征进行编码，并且同时测试早期融合与后期融合策略。分类结果展示在表4.5中。在这里，我们能够看出早期融合的独特作用。当视觉短语结构并未引入的时候，早期融合策略并没有体现出它的优势；然而，最好的分类结果来源于在几何视觉短语中同时编码两种不同的特征（SIFT和边缘SIFT）。只有早期融合能够帮助我们构建包含两种特征的视觉短

语（图4.17），从而对分类结果产生显著的提升。

4.4　基于边缘的空间加权

传统的词袋模型为所有特征设置同样的权值，然而这种假设在许多情况下并不成立，因为很多特征其实并非落在感兴趣区域内。由于这些兴趣区域外的特征通常会引入噪声，因此有必要减弱它们的权值以增强模型的表达能力。这等价于在图像平面上学习一个加权函数，或者检测图像中的显著性区域。在这里，我们遵循[193]的结果，认为图像中对比明显的位置有可能吸引更多的关注。根据这样的认识，我们提出一个基于边缘检测和高斯模糊（Gaussian decay）的空间加权算法。

4.4.1　边缘图像的模糊化

假设图像$\mathbf{I}$的大小为$W \times H$。边缘图像的定义仍然与(4-2)式相同。我们计算另外一个$W \times H$的矩阵$\mathbf{W}$用于空间加权：

$$\mathbf{W} = \left(w_{ij}\right)_{W \times H} \tag{4-18}$$

这里，w_{ij}是坐标(i, j)处的加权值，或者显著度，它由边缘强度累加而得：

$$w_{ij} = \sum_{i',j'} e_{i'j'} \times \exp\left\{-\sigma_e \left\|(i, j) - (i', j')\right\|_2\right\} \tag{4-19}$$

为了计算(i, j)处的加权值，我们需要在整个图像平面上枚举(i', j')。 σ_e是边缘平滑参数，而$\|\cdot\|_2$表示欧氏距离。随着σ_e的增加，边缘响应的作用范围逐渐减小。

加入空间加权的最大池化函数(4-9)变为：

$$\widetilde{\mathbf{F}^{\mathrm{GPP}}} = \max_{1 \le m \le M} \{w_m \times \mathbf{p}_m\} \tag{4-20}$$

其中，w_m表示在对应短语$\mathcal{G}_m$中心位置$\mathbf{l}_m$处的空间加权值。

4.4.2　加权算法的效果和讨论

我们在**Caltech101**数据集上测试不同的σ_e参数。关于GPP算法，参数设定与前述最优值一致。我们测试了不同的σ_e值：0、0.01、0.02、0.03、0.05和0.10，并将分类结果绘制于图4.14中。最优的参数值$\sigma_e = 0.05$将被应用于后续的实验中。

为了提供参数σ_e的效果的一个直观估计，我们将其对应的权值矩阵$\mathbf{W}$绘制为热度图。不同参数的结果如图4.18所示。当σ_e较小时，空间权值比较均匀地分散在全图的各个位置；随着σ_e的增加，空间权值开始变得越来越集中；当σ_e达到一

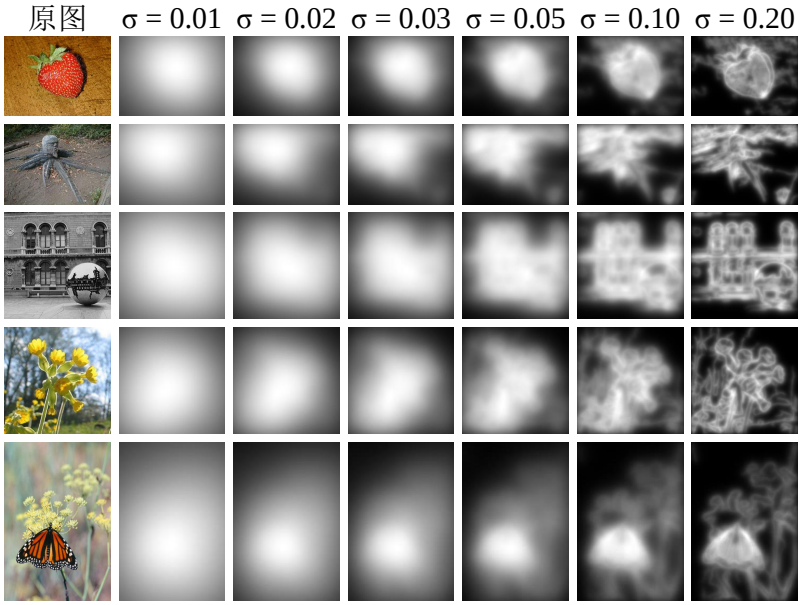


图 4.18　边缘平滑参数σ_e的效果：随着σ_e的增加，空间加权变得更加集中。这些样例图像来自于不同的数据集，所有选定的类都因为空间加权而获得了更好的分类精度。

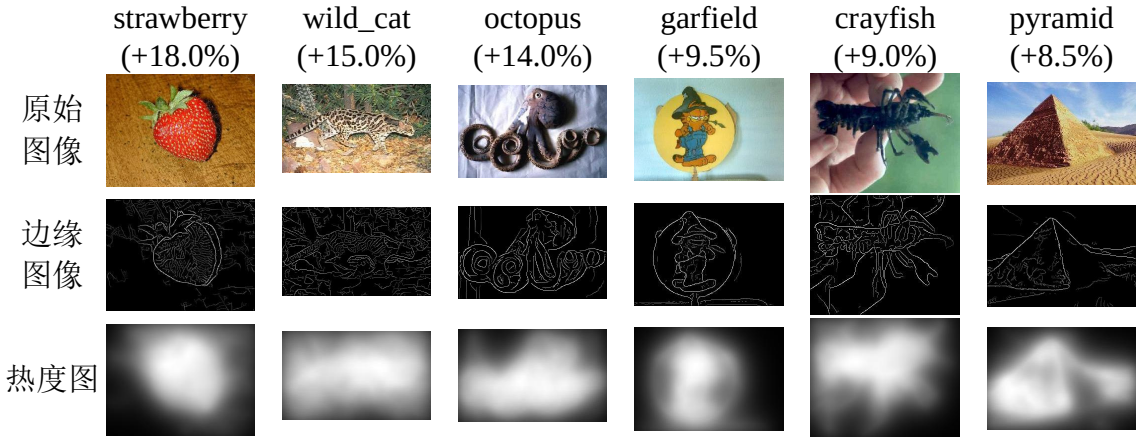


图 4.19　在空间加权后，准确率提升最多的4个类和降低最多的2个类。括号内的数字表示相应的准确率变化。

定大小（如σ_e = 0.20）后，由于权值的高度集中，热度图与边缘图像的形态已经非常相似。这样的加权方式事实上筛除了很多细节纹理信息，从而对于分类纹理敏感的类别（如动植物）非常不利。

注意到我们在GPP算法中也提到了视觉单词的加权（见第4.3.3.3节）。图4.9与图4.18存在一定的相似之处。它们的区别在于，GPP加强了局部区域内相似特征的联系，而空间加权则对于图像平面上特征的重要性进行了判断。这两种方法都有助于生成更具描述力的特征向量。

值得注意的是，检测显著性区域本身就是计算机视觉领域一个非常具有挑战性的问题。显然，基于边缘的加权算法并不能完全解决这个问题，然而它作为

一个简单有效的方案，取得了很好的分类结果。为了进一步分析，我们按类计算**Caltech101**数据集上的分类准确率，并且将一些经过空间加权后效果提升和下降比较明显的类集中在图4.19中。我们可以看到，空间加权方法在背景比较简单的情况下工作得较好，但是在显著性检测结果较差时，反而容易降低分类准确率。总体来说，在每一个数据集中，空间加权方法都对更多的类产生了正面作用，因此提高了平均分类精度。

4.4.3　计算复杂度

最后我们讨论计算复杂性的问题。注意到计算(4-19)式需要枚举图像上的每一个像素对（复杂度为$O\left(W^2H^2\right)$），这也让精确计算方法变得非常慢（对于一张300×300图像，单核2.0GHz的CPU需要大约30秒的计算时间）。为了加速，我们忽略那些欧氏距离超过50个像素的像素对。在上述最优设置（$\sigma_e = 0.05$）下，这种情况能够忽略的最大权值为$\exp\{-0.05\times50\}\approx0.08$。在这种近似下，我们的方法对于每张图像的处理时间大约为0.5秒。考虑到每张图像只需离线处理一次，这样的时间复杂度完全可以接受。

4.5　实验部分

在这一节里，我们提供我们的方法在若干图像分类数据集上的测试结果。

4.5.1　基本设置

为了将我们的算法与当前的先进水平进行比较，我们继承了前面章节中学习的最优配置。

- **边缘检测。** 我们使用Compass算子[149]进行边缘检测。遵循原文，我们将算子的半径参数（radius parameter）设置为4。
- **图像描述子。** 我们用VLFeat程序库[174]提取密集的SIFT特征和边缘SIFT特征。一张图像首先被重置大小，在保证其长宽比不变的情况下，将其长边重置为300个像素。密集采样的空间跨度和窗口大小将在每一个数据集内进行讨论。
- **视觉码本。** 我们用K-聚类算法训练码本。对于大部分数据集，码本大小为2048；对于对码本需求较大的Caltech256和PascalVOC2007数据集，码本大小分别为4096和8192。用于训练码本的描述子个数一般不超过2百万个。
- **特征编码和几何短语池化。** 我们使用LLC算法[9]对特征进行编码，并且用GPP强化局部编码。GPP的参数遵循4.3.3节：$K = 20$、$\sigma_w = 0.01$、

训练样本数	5	10	15	20	30
Lazebnik[10]	–	–	56.4	–	64.6
Yang[78]	–	–	67.0	–	73.2
Wang[9]	51.15	59.77	65.43	67.74	73.44
Boureau[190]	–	–	–	–	75.7
Bosch[18]	–	–	–	–	81.3
我们的方法	**61.90**	**71.75**	**76.03**	**78.53**	**82.45**
	±0.54	±0.60	±0.63	±0.39	±0.59

表 4.6　**Caltech101**数据集上的分类准确率（%）。

$r_1 = 5$以及$r_2 = 30$。

- **空间加权。** 在基于边缘的空间加权算法中，我们使用光滑参数$\sigma_e = 0.05$（见第4.4.2节）。
- **空间金字塔（SPM）和归一化。** 我们使用一个3层（$1 \times 1 + 2 \times 2 + 4 \times 4$）的SPM模型，用于加强空间特征编码的效果。随后，我们使用ℓ_2范数对整个长向量进行归一化。
- **分类过程。** 我们使用LibLINEAR[99]，一个可扩展的支持向量机（SVM）模型，进行训练和测试。 SVM的松弛参数始终设置为10。
- **精度计算。** 对于PascalVOC2007数据集，我们仿照[194]的方法计算平均准确率（mAP值）。对于其他数据集，我们在每类中选择一定数量的训练样本，并且报告所有类的平均分类准确率。我们重复10次随机的训练/测试数据划分，并且报告10次的平均准确率。

4.5.2　一般物体分类

Caltech101数据集[28]包含102类，9144张图像，包括一个“背景”类别。虽然这个数据集的图像对齐一般比较好，但是同一类的不同样本之间还是存在较大的类内变化。 SIFT特征的密集采样跨度为7，窗口大小为7；边缘SIFT特征的密集采样跨度为7，窗口大小为12。我们从每类中随机选取5、10、15、20和30个训练样本，并且将剩余的图像用于测试。表4.6对比了我们的方法和一些最近的方法。在所有的情形中，我们的方法都在LLC的基础上产生了很大的提高；在训练样本较少的情况下，精度提高甚至超过10%。

Caltech256数据集[29]包含257个类的30607张图像，包括一个“杂物”类别。这个数据集是对**Caltech101**的扩充，包括更多更具有挑战性的，对齐较差的图像。在许多情况下，我们都能观测到很强的类内变化和类间相似性。 SIFT特征的密集

训练样本数	5	15	30	45	60
Griffin[29]	–	28.3	34.1	–	–
Yang[78]	–	27.73	34.02	37.46	40.14
Gao[195]	–	29.77	35.67	38.61	40.30
Wang[9]	–	34.36	41.19	45.31	47.68
Bosch[18]	–	–	44.0	–	–
我们的方法	**26.12**	**36.35**	**45.07**	**48.02**	**50.33**
	±0.21	±0.31	±0.24	±0.25	±0.18

表 4.7　**Caltech256**数据集上的分类准确率（%）。

类名	[9]的结果	我们的结果	提升	类名	[9]的结果	我们的结果	提升
aeropl	67.47	72.29	**4.82**	*bicycl*	55.29	56.33	**1.04**
bird	40.68	45.41	**4.72**	*boat*	58.56	61.26	**2.71**
bottle	21.19	26.24	**5.05**	*bus*	44.10	53.77	**9.68**
car	69.43	73.56	**4.13**	*cat*	46.73	52.18	**5.44**
chair	51.50	54.19	**2.70**	*cow*	31.21	40.78	**9.58**
dining	35.06	47.40	**12.33**	*dog*	39.00	41.58	**2.57**
horse	72.41	74.38	**1.98**	*motorb*	53.98	57.52	**3.55**
person	79.18	83.02	**3.84**	*potted*	18.77	26.03	**7.80**
sheep	33.14	37.51	**4.37**	*sofa*	44.73	52.30	**7.57**
train	66.59	69.51	**2.92**	*tvmoni*	40.96	47.50	**6.53**
平均 ①	48.50	53.64	**5.14**				

表 4.8　**PascalVOC2007**数据集上的检索精度（%）。

① 这里列出的分类精度和文中报告的49.13和53.89略有不同。这是因为文中报告的（略高）是每一类在所有码本上取得的最好测试结果的平均，而这里所罗列的是某一次测试的结果。

采样跨度为5，窗口大小为5；边缘SIFT特征的密集采样跨度为8，窗口大小为8。我们从每类中随机选取5、15、30、45和60个训练样本，并且将剩余的图像用于测试。表4.7对比了我们的方法和一些最近的方法。再一次地，我们提出的系统稳定地超过了LLC，达到更好的分类精度。

作为一个比赛数据集，**PascalVOC2007**数据集[194]包含了可能含有20类物体的9963张图像。这是一个非常具有挑战性的数据集，因为其中大部分的物体形变非常严重，而且许多情况下物体只占据图片的一小块区域。我们的任务是训练20个独立的分类器，并且用它们来预测每一张测试图片中出现每一类物体的可能性。根据**PascalVOC**的标准测试方法，我们报告算法的mAP值（一种用于评价检索系统的标准）。

训练样本数	1	5	10	20	26
Lazebnik[196]	–	–	–	–	90.4% ①
Larlus[197]	–	–	–	–	90.61% ②
Wang[9]	54.41	77.98	83.38	86.33	87.54
我们的方法	**64.25**	**80.19**	**85.30**	**88.93**	**90.83**
	±6.46	±1.72	±1.97	±1.00	±1.34

表 4.9　**Butterfly-7**数据集上的分类准确率（%）。

① 这篇文章中使用的基于部件的分类方法，其复杂度远高于我们的方法。

② 这是该文章中报告的最好的一次训练/测试划分的运行结果，我们的算法在最好的一次划分中取得了93.31%的准确率。

训练样本数	5	10	20	30	60
Nilsback[32]	–	–	–	–	81.3%
Gehler[198]	–	–	–	–	85.5% ①
Wang[9]	69.52	76.98	82.43	84.94	88.24
我们的方法	**72.08**	**79.39**	**84.47**	**86.94**	**91.56**
	±1.60	±0.87	±1.16	±0.67	±1.61

表 4.10　Oxford **Flower-17**数据集的分类准确率（%）。

① 这是在一个固定的训练/数据划分上报告的结果。在该分割上，我们的方法得到更高的91.43%准确率。

SIFT特征的密集采样跨度为6，窗口大小为4；边缘SIFT特征的密集采样跨度为7，窗口大小为12。按照每类物体测试的准确率列在表4.8中。我们的系统在20类中的每一类都超过了LLC的结果，平均精度53.89%也明显高于LLC的49.13%。

4.5.3　特定物体分类

Butterfly-7数据集[196]包含7种蝴蝶，619张图像。在每一类中，我们都能找到一些具有挑战性的样本，如物体占据图像区域非常小，或者有多个物体出现在一张图像中。我们使用密集采样的OpponentSIFT特征[70]，空间跨度为7，窗口大小为12。鉴于第4.2.4节讨论的原因，我们在这里不使用边缘SIFT特征。我们从每类中随机选取1、5、10、20和26个（数据集指定数目）训练样本，并且将剩余的图像用于测试。分类结果如表4.9所示。

Oxford **Flower-17**数据集[32]包含17种花，每类80张图片。每一类的图像样本通常在物体数量、尺度、视角和光照等方面存在较大差距。我们使用密集采样的OpponentSIFT特征[70]，空间跨度为7，窗口大小为12。鉴于第4.2.4节讨论的

训练样本数	10	20	30	50	100
Lazebnik[10]	–	–	–	–	81.4
Li[199]	–	–	–	–	80.9
Gao[195]	–	–	–	–	83.68
Wang[9]	66.97	72.44	75.78	78.84	82.34
我们的方法	**70.67**	**76.12**	**78.74**	**81.72**	**85.13**
	±0.46	±0.73	±0.88	±0.48	±0.72

表 4.11　**Scene-15**数据集的分类准确率（%）。

训练样本数	5	10	20	40	80
Quattoni[189]	–	–	–	–	26.0
Li[199]	–	–	–	–	37.6
Bo[200]	–	–	–	–	41.8
Wang[9]	19.31	25.61	31.34	37.01	43.10
我们的方法	**21.11**	**27.64**	**34.22**	**40.56**	**46.38**
	±0.49	±0.69	±0.45	±0.60	±0.75

表 4.12　MIT **Indoor-67**数据集的分类准确率（%）。

原因，我们在这里不使用边缘SIFT特征。我们从每类中随机选取5、10、20、30和60个（数据集指定数目）训练样本，并且将剩余的图像用于测试。分类结果如表4.10所示。

4.5.4　场景识别

Scene-15数据集[10]包含15种场景，4485张图片。所有的类别都是室外场景，所有的图像都是灰度图像。这是早期最常用的场景识别数据集之一。 SIFT特征的密集采样跨度为5，窗口大小为5；边缘SIFT特征的密集采样跨度为8，窗口大小为8。我们从每类中随机选取10、20、30、50和100个训练样本，并且将剩余的图像用于测试。表4.11陈列了实验结果。

MIT **Indoor-67**数据集[189]包含67种室内场景，15620张图像。这是一个非常常用的室内场景识别任务，由于类别数量的增加，不少类之间的差别已经相对比较小。 SIFT特征的密集采样跨度为6，窗口大小为4。由于室内场景的边缘信息通常非常接近，我们没有采用边缘SIFT特征。我们从每类中随机选取5、10、20、40和80个训练样本，并且将剩余的图像用于测试。表4.12陈列了实验结果。

4.5.5　讨论

在前面的实验中，我们已经说明：包含三个新模块（抽取互补特征、几何短语池化、基于边缘的空间加权）的系统产生了很好的分类结果。此处，我们在**Caltech101**数据集上进一步分析我们的实验结果。我们使用每类30张训练图像，在此设定上我们取得了9%的精度提升（12%相对提升）。

- **抽取互补特征**对于分类准确率有大约5%的提升。互补性不仅体现在SIFT特征和边缘SIFT特征所捕捉的视觉线索的不同，也体现在不同的空间跨度和窗口大小能够捕捉不同的信息。可以看到，不同窗口大小的特征也被广泛应用在其他分类方法中[9][86]。
- **几何短语池化**（GPP）对于分类准确率有大约4%的提升。 GPP对于提供了一种简单有效的方式，以编码一些局部特征的组合。 GPP算法本身的时间复杂度很低，由于对特征稀疏性的影响，将会使得在线分类模块花费更多的时间。
- **基于边缘的空间加权**对于分类准确率有大约2%的提升。虽然我们在不同的类别上同时观测到了精度的上升和下降，但是平均精度可以有稳定的提升。

总体来说，我们提出的三个模块能够独立地应用于分类问题，产生稳定的效果提升。由于边际效应，整个系统产生的精度提升或许会小于三个模块各自的精度提升之和，然而我们说明了这些模块依然能够互相配合，提升分类效果。

4.6　本章小结

本章提出了三个新的应用于分类问题的模块，即抽取互补特征、几何短语池化以及基于边缘的空间加权。这三种方法能够独立地对分类问题产生帮助，并且能够在一个完整的框架中互相配合，使得算法的分类结果达到先进水平。其中，抽取互补特征和几何短语池化算法能够建立局部的特征组，并且为其产生紧凑的特征表示。它们的成功说明，局部特征之间存在很强的相关性，如果能够显式地捕捉这些相关性，就能够提供更强的特征表示。

同时，我们的算法还有一些不足之处。例如，过于简单的几何短语结构无法根据物体的大小调整视觉短语的形态；而简单的基于边缘的空间加权在很多时候也无法找到真正值得注意的物体位置。对于这些问题的研究将留待后续工作。

第 5 章　图像分类：局部特征的优化组合

5.1　研究动机

传统的视觉词袋模型主要采用朴素的空间切分，如空间金字塔（Spatial Pyramid Matching，SPM）模型，将空间信息编码到特征向量中。然而对于一些特定的问题，如细粒度物体识别（fine-grained object recognition）和场景分类（scene classification），简单的空间切分并不能产生很好的效果，因为它们往往无法捕捉特定任务中的特定概念（如细粒度物体的部件信息）。

在这一章里，我们主要讨论局部特征的优化组合对于分类问题产生的帮助。我们针对两类特殊的图像分类问题，即细粒度物体识别问题和场景分类问题，分别设计两种不同的局部特征组合方式。对于细粒度分类问题，我们提出**层次化部件匹配**（Hierarchical Part Matching，HPM）算法，通过对物体部件的检测以及基于部件的中层和高层语义概念匹配，达到了显著的提升效果。对于场景分类问题，我们观察到朝向信息对于分类的重要性，因而提出**朝向金字塔匹配**（Orientational Pyramid Matching，OPM）算法，以提供朝向信息作为空间信息的补充。这两部分的实验结果都印证了我们的研究动机：通过优化的特征组合方式，它们在相应的分类任务上都取得了良好的效果。

与本章相关的出版物为[192] [187] [201]。

5.2　朴素的空间切分：空间金字塔匹配

我们首先简单回顾应用于分类的特征组合算法，尤其是池化算法。同时我们也将SPM算法推广至更一般的情形，说明即使简单的扩展也能起到良好的效果。

5.2.1　特征组合与指数子集

令$\mathcal{D}$表示描述子集合：$\mathcal{D} = \{(\mathbf{d}_1, \mathbf{l}_1), (\mathbf{d}_2, \mathbf{l}_2), \ldots, (\mathbf{d}_M, \mathbf{l}_M)\}$；而$\mathcal{W}$表示编码后的特征集合$\mathcal{W} = \{(\mathbf{w}_1, \mathbf{l}_1), (\mathbf{w}_2, \mathbf{l}_2), \ldots, (\mathbf{w}_M, \mathbf{l}_M)\}$。其中，$\mathbf{d}_m$、$\mathbf{w}_m$和$\mathbf{l}_m$分别表示第$m$个描述子（特征）的描述向量、特征向量以及位置。有时，我们需要显式地表达描述子（特征）对应的图像区域$\mathcal{R}_m$，它一般由$\mathbf{l}_m$和相应的几何性质确定。特征编码的组合方法需要将这个集合压缩为一个单独的长向量，作为图像的表示向量。其

中，最常见的手段是池化（pooling），即在M个向量上定义一种保维（dimension-preserving）运算。典型的例子如最大池化（max-pooling）：$\mathbf{F} = \max_m \mathbf{w}_m$；以及平均池化（average-pooling）：$\mathbf{F} = \frac{1}{M}\sum_m \mathbf{w}_m$。然而，全局的池化通常会忽略重要的空间信息，为此我们可以采用特征分组的方式对它们进行空间相关的组合。

显式地，整个特征集合$\mathcal{W}$的索引集（index set）为$\mathcal{J} = \{1, 2, \ldots, M\}$。特征分组就是定义一个包含$\mathcal{J}$的$S$个子集的子集组（subset group），记为$\{\mathcal{J}_1, \mathcal{J}_2, \ldots, \mathcal{J}_S\}$，然后在每个子集组内分别对特征进行池化，得到$S$个独立的池化向量$\{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_S\}$。最后，这些池化的向量被拼接为一个长向量$\mathbf{F}$，作为词袋模型的输出。

5.2.2　标准的金字塔匹配

不同的特征分组方法将会产生不同的效果。经典的空间金字塔匹配（Spatial Pyramid Matching，SPM）[10]算法首先定义一个整数L，然后将图像分为L个层次，并且在每一层上对特征进行层次化的切分和组合。

令$\mathcal{P}$表示图像$\mathbf{I}$的像素集合。对于第0层，我们定义唯一一个池化箱（pooling bin）：$\mathcal{P}_{0,0} = \mathcal{P}$。随后的定义是递归进行的。对于$l \geqslant 0$和$0 \leqslant t < 4^l$，我们将第$l$层的第$t$个池化箱分为四份，作为第$l+1$层的池化箱，即：$\mathcal{P}_{l+1,4t} = \mathcal{P}_{l,t}^{\mathrm{UL}}$、$\mathcal{P}_{l+1,4t+1} = \mathcal{P}_{l,t}^{\mathrm{UR}}$、$\mathcal{P}_{l+1,4t+2} = \mathcal{P}_{l,t}^{\mathrm{LL}}$以及$\mathcal{P}_{l+1,4t+3} = \mathcal{P}_{l,t}^{\mathrm{LR}}$。这里，$\mathcal{P}_{l,t}^{\mathrm{UL}}$、$\mathcal{P}_{l,t}^{\mathrm{UR}}$、$\mathcal{P}_{l,t}^{\mathrm{LL}}$和$\mathcal{P}_{l,t}^{\mathrm{LR}}$分别表示$\mathcal{P}_{l,t}$在均匀切分为$2 \times 2$区域后的左上角、右上角、左下角和右下角。很容易知道，第l层有$2^l \times 2^l$个池化箱，每一个的大小都是$\left(W/2^l\right) \times \left(H/2^l\right)$。根据池化箱，我们就可以相应地定义指数子集：$\mathcal{J}_{l,t} = \{m \mid 1 \leqslant m \leqslant M \wedge \mathbf{l}_m \in \mathcal{P}_{l,t}\}$。一个$L$层模型中总共的池化箱个数（指数子集个数）为：$\sum_{l=0}^{L-1}\left(2^l\right)^2$。

5.2.3　推广的规则匹配

我们将SPM进行简单的推广，提出**推广的规则空间池化**（Generalized Regular Spatial Pooling，GRSP）算法。我们仍然遵循SPM对于空间划分的定义，但是允许每层的池化箱数量发生变化，从而定义更加灵活的特征组合算法。

首先，与SPM算法保持一致，我们仍然假设模型有L层，且第l层的池化箱的大小为$\left(W/2^l\right) \times \left(H/2^l\right)$。为此，我们定义一个序列$(s_0 = 1, s_1, s_2, \ldots, s_{L-1})$，表示第$l$层有$s_l \times s_l$个相同大小的池化箱。在第$l$层，我们首先在图像的左上角放置一个$\left(W/2^l\right) \times \left(H/2^l\right)$大小的池化箱，然后将这个池化箱沿着图像的长宽方向，从左上角到右下角移动，并且保证它每次沿着两个方向移动的跨度是一致的 ①。显然，

① 例如，图像的宽和高分别为300和200像素，而池化箱的宽和高分别为120和80像素。如果在某层$s = 4$，

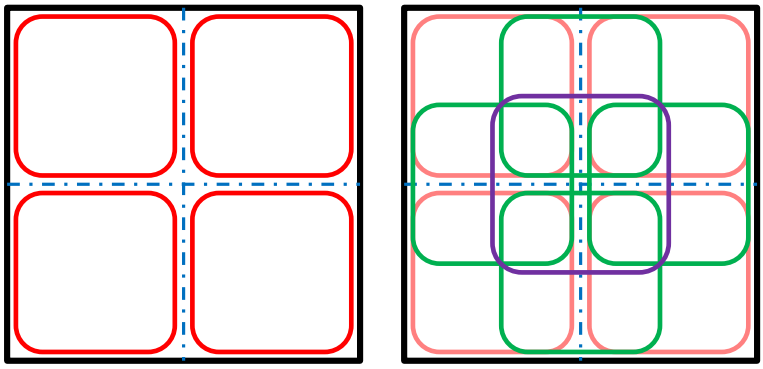


图 5.1　在第1层上，原始池化（左）和更加密集池化（右）的例子。池化箱的大小为$\frac{W}{2} \times \frac{H}{2}$)，当$s_1 = 3$时，每个池化箱和相邻的箱共享一半像素点。

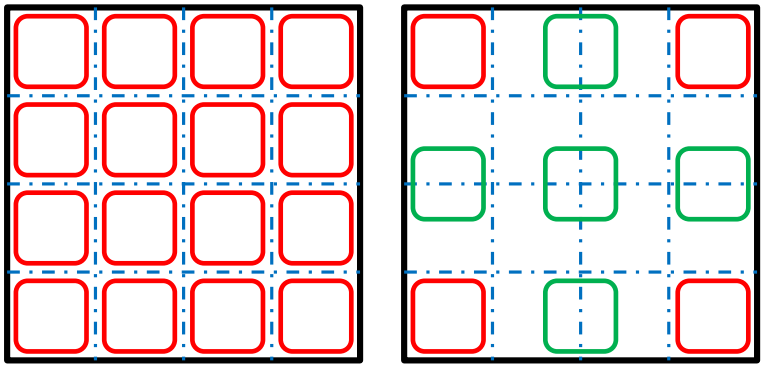


图 5.2　在第2层上，原始池化（左）和更加稀疏池化（右）的例子。池化箱的大小为$\frac{W}{4} \times \frac{H}{4}$)，当$s_2 = 3$时，某些像素点并没有被任何一个池化箱覆盖。

当$s_l = 2^l$时，第l层的池化算法就退化为与SPM相同的情形，否则我们就会得到更密集（$s_l > 2^l$）或者更稀疏（$s_l < 2^l$）的池化箱。图5.1展示了第1层上更加密集的池化（$s_1 = 3$），而图5.2展示了第2层上更加稀疏的池化（$s_2 = 3$）。更加密集的池化可能使得某些像素在同一层中被超过一个池化箱覆盖，而更加稀疏的池化可能使得某些像素在某一层中不属于任何池化箱。

根据上述定义的池化箱，我们就可以得到相应的子集组。显然，在GRSP中，指数子集的数量为$\sum_{l=0}^{L-1} s_l^2$。

5.2.4　实验部分

5.2.4.1　数据集和基本设置

我们在四个常用的数据集上报告分类准确率，包括UIUC **Sport-8**数据集[202]

那么每次在宽度和高度方向上的跨度就分别为60和40像素。

情形编号	编码算法	s_l 第0层	第1层	第2层	特征维度	Sport-8 精度 (%)	Scene-15 精度 (%)	Indoor-67 精度 (%)	Caltech101 精度 (%)
1	LLC	1×1	2×2	3×3	28K	87.28	81.34	43.21	73.24
2	LLC	1×1	2×2	4×4	42K	87.03	81.66	43.55	74.47
3	LLC	1×1	2×2	6×6	82K	86.73	81.76	44.63	75.96
4	LLC	1×1	2×2	8×8	138K	86.46	81.27	44.40	76.18
5	LLC	1×1	3×3	3×3	38K	87.60	81.89	43.22	75.43
6	LLC	1×1	3×3	4×4	52K	87.44	81.83	43.17	75.66
7	LLC	1×1	3×3	6×6	92K	87.09	81.90	45.15	76.70
8	LLC	1×1	3×3	8×8	148K	86.78	81.49	44.86	76.68
9	LLC	1×1	4×4	3×3	52K	87.58	81.48	44.04	75.61
10	LLC	1×1	4×4	4×4	66K	87.56	81.57	44.22	75.96
11	LLC	1×1	4×4	6×6	106K	87.18	81.67	45.07	76.55
12	LLC	1×1	4×4	8×8	162K	86.98	81.43	44.99	76.77
13	IFV	1×1	2×2	–	200K	90.82	87.54	61.22	80.73
14	IFV	1×1	3×3	–	400K	91.38	87.79	62.55	81.86
15	IFV	1×1	4×4	–	680K	91.16	87.75	62.57	82.04

表 5.1　四个数据集上不同模型和参数的分类准确率（%）。

（8种体育场景，1579张图像）；**Scene-15**数据集[10]（15种室外场景，4485张图像）；MIT **Indoor-67**数据集[189]（67种室外场景，15620张图像）；以及**Caltech101**数据集[28]（102个一般物体类，9144张图像）。其中，我们从每类随机选取70、100、80和30张图像用于训练，而将剩余图像用于测试。

我们采用局部限制的线性编码（Locality-constrained Linear Coding，LLC）和改进的Fisher向量（Improved Fisher Vectors，IFV）算法。基础实验设定分别遵循LLC编码[9]和IFV编码[7]的原始论文。一张图像首先被重置大小，在保证其长宽比不变的情况下，将其长边重置为600个像素。我们用VLFeat[174]，一个通用的计算特征代码库，提取密集的RootSIFT特征[118]。密集采样中，相邻特征的跨度为10像素，而特征的滑动窗口大小为16像素。在使用IFV编码时，SIFT特征被PCA降维至80维。我们分别采用K-Means和GMM模型，训练用于LLC和IFV的码本，用于训练的描述子大约有2百万个。在编码过程中，特征向量在每个池化箱中独立地进行归一化[94]。最后被送入LibLINEAR[99]（一个通用的线性SVM模型）进行训练和测试。我们报告的分类准确率，是算法在所有类的测试图像上的平均分类精度（10次随机训练和测试切分后取平均）。

5.2.4.2　模型和参数

本节将观察不同的模型和参数对GRSP算法的影响。实验结果见表5.1。

首先，我们对比LLC编码[9]下的不同实验结果。可以观察到，当第1层的池化箱数量从2×2增加到3×3时，分类的准确率通常能够得到明显的提升（见对比组$(1,5)$、$(2,6)$、$(3,7)$和$(4,8)$）。然而，当池化箱的数量进一步从3×3增加到4×4时，准确率的提升变得非常有限（见对比组$(5,9)$、$(6,10)$、$(7,11)$和$(8,12)$）。这表明，适度增加池化箱的数量能够捕捉更丰富的图像信息，但是过于密集的池化也会导致冗余信息增加。类似的情况也出现在第2层（见对比组$(1,2,3,4)$、$(5,6,7,8)$和$(9,10,11,12)$）：当池化箱的数量从4×4增加到6×6时，我们能够从增加图像信息中获得好处，但是过多的池化箱（8×8）同样会造成精度下降。

在IFV编码[7]上，我们观察到了类似的现象。此处，遵循原始论文，我们只采用两层的池化结构，其中第0层依然为全局池化。当第1层的池化箱从2×2增加到3×3时，分类准确率明显提高；但是进一步增加到4×4时产生的效果十分有限。

同时，我们观察池化箱的个数与数据集的类别个数的关系。在UIUC **Sport-8**数据集里，由于类别个数相对较少，使用长特征向量更容易造成过拟合。然而随着类别个数的增加，长向量编码的优势就逐渐体现出来。以LLC编码为例（IFV编码的结果非常类似），在UIUC **Sport-8**数据集上，第2层上使用3×3池化箱产生的结果最好；然而在**Caltech101**数据集上，密集的池化（第2层使用8×8池化箱）能够产生更好的分类效果。这就说明，在更大的数据集（如**Caltech256**[29]、**SUN-397**[31]和**ImageNet**[30]）上，可能有必要使用更加密集的池化。

作为总结，我们LLC编码后使用3层GRSP模型，每层分别为1×1、3×3、6×6池化箱；除非对于UIUC**Sport-8**数据集，我们在第2层使用3×3池化箱。这将产生92K维向量（在UIUC **Sport-8**数据集上，维度为38K），大约是原始SPM产生特征向量（1×1、2×2、4×4池化箱，42K维）维度的两倍。对于IFV编码，我们在2层GRSP模型中使用1×1和3×3池化箱，得到400K维特征向量，其维度恰好是原始SPM产生的特征向量（1×1和2×2池化箱，200K维）维度的两倍。

5.2.4.3　与其他方法对比

最后，我们在这些分类数据集上，对比我们的方法和其他方法的结果。为了让对比更加公平，我们比较那些只使用灰度SIFT特征，并且采用了朴素SPM模型的方法。对比结果如表5.2所示。在相对较好的基准上，GRSP方法仍然有效地提升了分类效果，使得我们报告的分类准确率达到了较高的水平。

算法	UIUC Sport-8	Scene-15	MIT Indoor-67	Caltech101
Yang[78]	–	80.4	–	73.2
Bo[203]	–	–	51.2	82.5
Jia[87]	–	–	–	75.3
Xie[71]	88.17 ± 0.78	83.77 ± 0.69	46.38 ± 0.75	78.14 ± 0.80
Kobayashi[82]	90.42	85.63	58.91	–
Wang (LLC)[9]	87.10 ± 0.82	81.66 ± 0.36	43.55 ± 0.63	74.47 ± 0.91
我们的方法 (LLC + GRSP)	**87.60** ± 0.73	**81.89** ± 0.50	**45.15** ± 0.46	**76.70** ± 0.79
Perronin (IFV)[7]	90.82 ± 0.92	87.54 ± 0.58	61.22 ± 0.65	80.73 ± 0.82
我们的方法 (IFV + GRSP)	**91.38** ± 0.86	**87.79** ± 0.59	**62.55** ± 0.45	**81.86** ± 0.94

表 5.2　我们的方法与其他方法的分类精度（%）对比。

5.2.5　结论

本节主要介绍了SPM方法，并且提出了一种简单的改进，即GRSP算法。我们可以看出，GRSP算法虽然简单，但是却能够稳定地提升分类效果，因为它对图像空间信息的编码更加充分合理。

在下面的章节中，我们将研究两类特殊的分类问题，即细粒度物体识别和场景识别，提出两种更具针对性的特征组合（池化）算法。

5.3　细粒度分类：层次化部件匹配

5.3.1　问题综述

与一般的图像分类问题不同，细粒度物体分类（fine-grained object recognition）问题一般需要区分一些语义非常接近的类，例如大量的*动物*[34]、*植物*[33]或者*人造物品*[36]。由于这些物体的区别往往体现在一些细节方面（例如某些局部的纹理或者形状），传统缺乏对齐的特征组合方式往往无法取得良好的效果。

本节提出一种新颖的分类框架，称为**层次化部件匹配**（Hierarchical Part Matching，HPM），用于细粒度物体识别。我们充分利用了物体上的弱标注，将其扩展为一些基本区域，并且利用这些区域来实现更好的部件对齐和语义理解。HPM模型的流程如图5.3所示。首先，我们利用弱标注信息，对图像的部件切分进行推理，以形成一些具有基本语义的图像区域；其次，我们提出层次化结构学习（Hierarchical Structure Learning，HSL）算法构建中层结构，以捕捉更多的语义部件；最后，我们应用几何短语池化（Geometric Phrase Pooling，GPP，见第4章）算法对局部特征进行强化编码。将上述模块结合起来，能够得到一个强力的细粒度物体识别模型，并取得良好的分类效果。本节的主要贡献集中在对细粒度分类

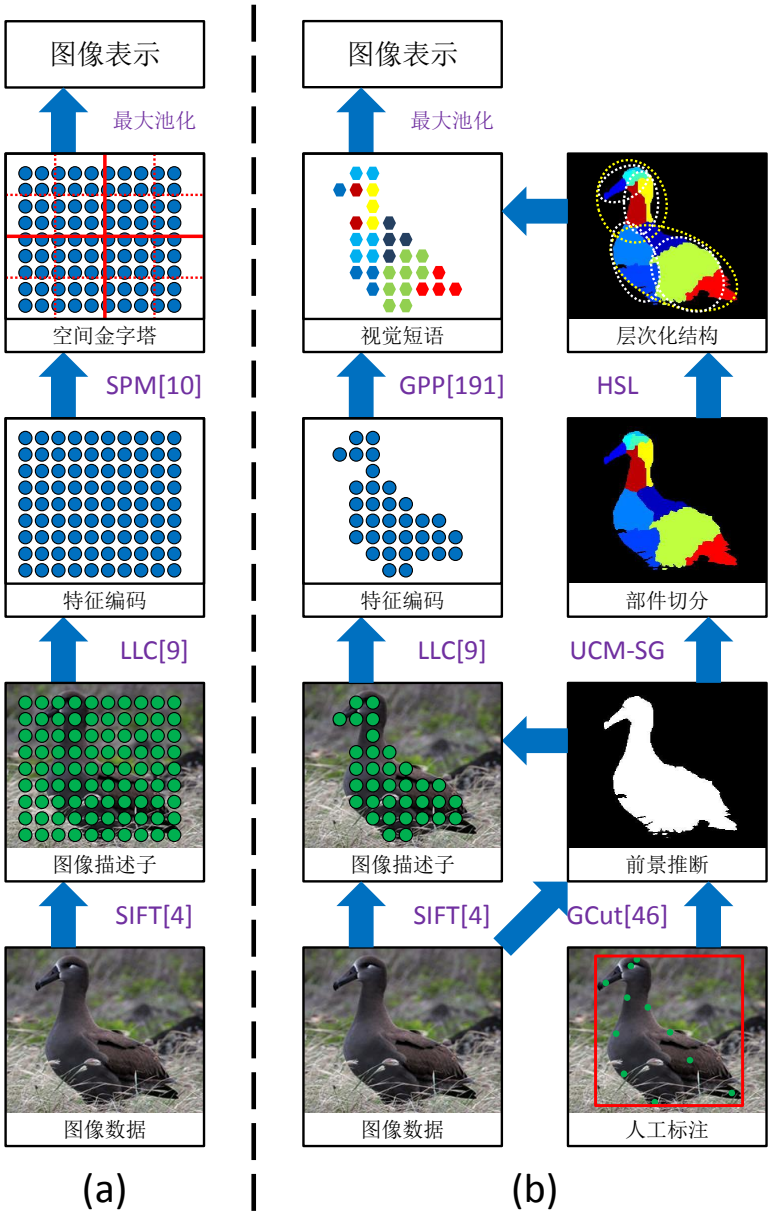


图 5.3　传统的SPM框架(a)和我们提出的HPM框架(b)的对比。

问题提供的线索。我们将说明：细粒度分类问题的关键在于物体部件的检测和切分；后续研究应当在这一方面投入更多的精力。

5.3.2　细粒度分类数据集

我们使用Caltech-UCSD **Birds-200**-2011（CUB-200-2011）数据集[34]。这是一个专门为细粒度物体识别设计的数据集，包含200个生物学鸟类分类，以及11788张图像（大约每类60张）。此外，该数据集还提供了每张图像的一个包围框（包含一只鸟）以及至多15个部件的标注点，对应于15个可能出现的部件。数据集的一些样例如图5.4所示。在这些非常具有挑战性的例子上，即使是经过训练

图 5.4　Caltech-UCSD **Birds-200**-2011数据集[34]上的样例图像。上部：来自同一类（#001：*Black-footed Albatross*）的四张图像，说明类内物体的较大差别；下部：来自四个不同类的四张图像，说明类间物体可能的相似性。

的人类通常也很难做出准确的判断。该数据集的主要优势在于，提供了一些弱标注信息（包围框和标注点），这将帮助我们的算法取得更好的分类效果。

5.3.3　物体部件的切分

5.3.3.1　前景推断

细粒度分类问题的一个显著特点是，几乎所有图像都有非常类似的背景。以Caltech-UCSD **Bird-200**-2011数据集为例，几乎所有的类别中都出现了水面、树木和草地。这些背景将会引入不稳定的噪声特征，因此我们考虑将它们筛除，以提升模型的表达能力。

我们使用抓取切割（Grab-Cut）算法进行前景推断[46]。Grab-Cut是一个迭代算法，其初始的掩图（mask）由图像中物体的包围框和部件标注点生成。准确地说，我们将物体包围框外的像素设定为**绝对背景**，包围框内部的像素设定为**可能前景**，而所有部件标注点周围一个小范围（通常为2–5像素）内的像素设定为**绝对前景**。Grub-Cut算法经过大约10轮迭代，就能够收敛，从而产生最后的结果。图5.5展现了完整的前景推断过程。

5.3.3.2　能量函数和部件切分

在获得前景区域后，我们自然希望将其进一步分割为若干包含基础语义的部件，以辅助特征组合算法。为此，我们寻找可能的边界位置。我们计算超度量轮廓图（Ultrametric Contour Map，UCM）[204]，它将在图像上生成一系列闭合的轮廓（contours），位于轮廓上的像素都被赋予一个非零值，表示该像素位于物体边

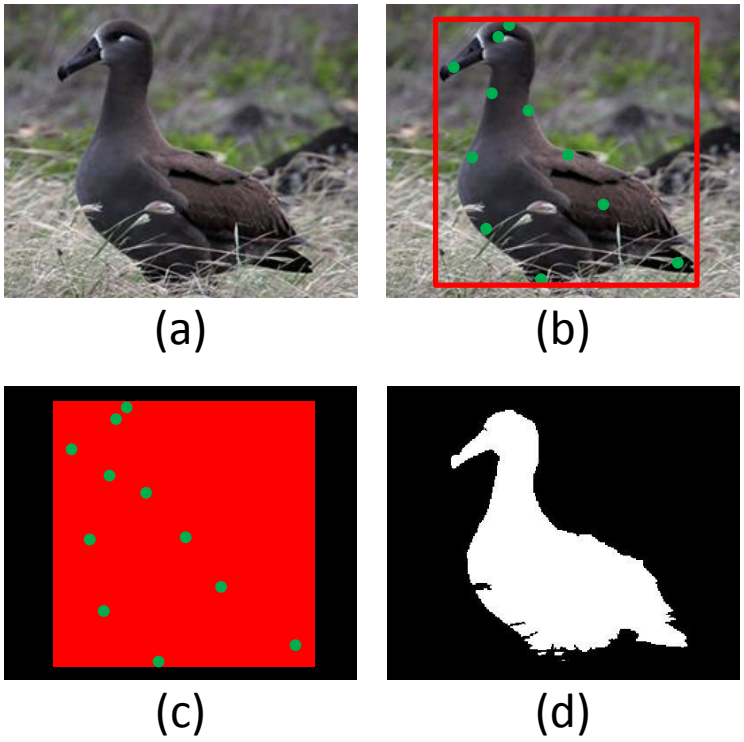


图 5.5　前景推断过程。 (a)原始图像； (b)红色的物体包围框，以及绿色的部件标注点；(c)Grab-Cut算法产生的初始掩图，其中黑色、红色和绿色区域分别表示绝对背景、可能前景和绝对前景区域； (d)迭代10轮后的前景切分结果。

界上的概率。记产生的轮廓图为$\mathbf{U}$（与图像大小相同的矩阵）：

$$\mathbf{U} = \left(u_{ij}\right)_{W \times H} \tag{5-1}$$

这里，u_{ij}表示(i, j)位置的边界强度（boundary intensity）。

基于UCM，我们就可以构建一个有向图$\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$，其中节点集$\mathcal{V} = \left\{v_{ij}\right\}$包含所有的像素点，而边集$\mathcal{E}$由相邻像素对应的节点连边组成。

$$\mathcal{E} = \left\{\left(v_{ij} \to v_{i'j'}\right) \mid \left|i - i'\right| + \left|j - j'\right| = 1\right\} \tag{5-2}$$

每条边的权值由它的尾节点的边界强度定义：

$$w\left(v_{ij} \to v_{i'j'}\right) = u_{i'j'} + \lambda \tag{5-3}$$

这里，λ被称为步长惩罚数（step penalty），用以考虑几何距离对分割造成的影响。图5.6表示在图像的一个局部构建的，$\mathcal{G}$的一个子图。

在图$\mathcal{G}$构建完成后，就可以利用部件标注点作为源（source），计算它们到每个像素的最短距离。使用改进的Dijkstra最短路算法，能够在$O(LP\log(P))$时间内完成计算，其中L是标注点的个数，而$P = W \times H$表示图像的像素数量。将第l个标

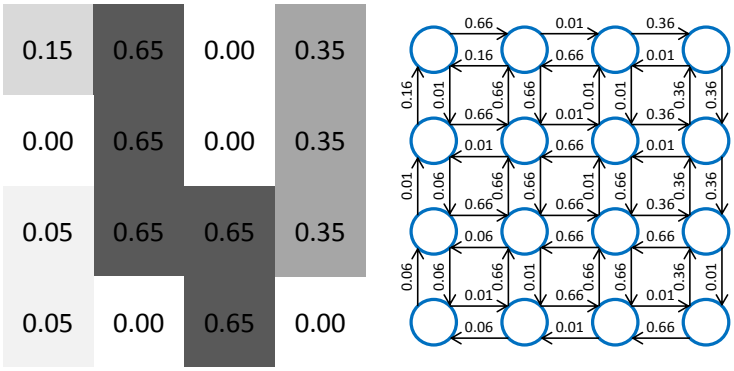


图 5.6　图$\mathcal{G}$的构建过程。左：UCM上的一个小区块，其中的每一个正方形表示一个像素点，其中的数字表示像素点上的边界强度。右：在这个区块上构建的$\mathcal{G}$子图，其中惩罚数λ定为0.01。

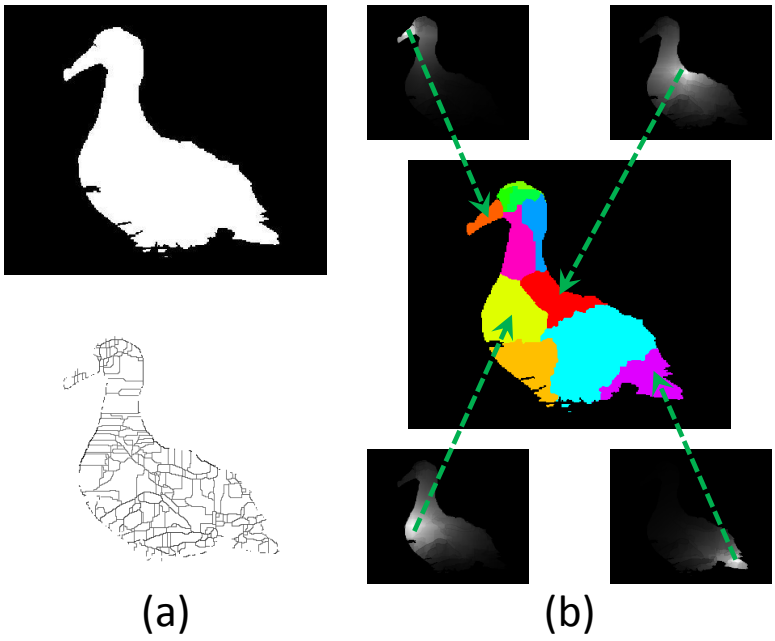


图 5.7　部件切分的过程。 (a)上下图分别为前景推断结果和相应的UCM（深色像素表示较大的边界强度）； (b)由像素到*beak*（左上角）、*breast*（左下角）、*back*（右上角和*tail*（右下角）的距离生成的热度图（heatmap）。对于所有的距离取最小值，就能够得到每个像素的分配结果（即中间展示的分割结果）。

注点到节点v_{ij}的距离记为$\{d(p_l, v_{ij})\}$。为了后续方便，我们将$\{d(0, v_{ij})\}$定义为背景距离（background distance）：当v_{ij}确实为背景像素时，它的取值为零；否则，它的取值为正无穷。

部件分割过程的目标是将每个前景上的像素分配给一个部件。将任意一个分

配方案记为**S**：

$$\mathbf{S} = \left(s_{ij}\right)_{W\times H} \tag{5-4}$$

其中，s_{ij}表示像素（节点）v_{ij}被分配到的部件编号，满足$0 \leqslant s_{ij} \leqslant L$。这里，$s_{ij} = 0$表示$v_{ij}$被分配到背景区域。为了量化分割的质量，我们定义一个能量函数，它由每个像素点到它被分配部件的标注位置的距离求和而得：

$$f(\mathbf{S}) = \sum_{i,j} d\left(p_{s_{ij}}, v_{ij}\right) \tag{5-5}$$

方程(5-5)的最小值就对应最优的分割方案。显然，(5-5)的优化过程可以转化为一系列独立的最小化过程：

$$s_{ij}^{\star} = \arg\min_{0\leqslant l\leqslant L} d\left(p_l, a_{ij}\right) \tag{5-6}$$

直观的部件分割过程如图5.7所示。

5.3.4　层次化结构学习

将全部像素的集合记为$\mathcal{P}$。前景推断和部件切分算法将$\mathcal{P}$分为至多L个前景区域和一个背景区域：

$$\mathcal{P} = \bigcup_{l=0}^{L} \mathcal{P}_l \tag{5-7}$$

这里，$\mathcal{P}_l$（$l > 0$）表示第l个分割的部件区域，而$\mathcal{P}_0$表示背景区域。分割区域都是互斥的：对于$l_1 \neq l_2$有$\mathcal{P}_{l_1} \cap \mathcal{P}_{l_2} = \varnothing$。

值得注意的是，分割出的区域都表示一些基本的部件，如*nape*、*left eye*、*right leg*，等等。从语义角度分析，可能存在一些中层或者高层的概念，它们由这些基本部件组成，并且包含更加丰富的语义信息。例如，视觉概念**eyes**可能由*left eye*和*right eye*组成，而概念**head**由*forehead*、*crown*、*beak*和**eyes**组成。为了将基本的部件组合在一起，我们寻找几何位置接近且表观特征类似的部件。为此，我们需要定义任意两个部件之间的几何距离（geometric distance）和特征距离（feature distance）。

我们回到描述子的集合$\{\mathbf{d}_m, \mathcal{R}_m\}$（见第5.2.1节）。我们利用标注的部件点计算几何距离，并且使用分割区域内的描述子计算特征距离。显式地，对于一张图像，如果集合$\mathcal{P}_{l_1}$和$\mathcal{P}_{l_2}$都非空，那么它们的几何距离定义为：

$$\mathrm{dist_g}(\mathcal{P}_{l_1}, \mathcal{P}_{l_2}) = \left[\left(p_{l_1}^{\mathrm{X}} - p_{l_2}^{\mathrm{X}}\right)^2 + \left(p_{l_1}^{\mathrm{Y}} - p_{l_2}^{\mathrm{Y}}\right)^2\right]^{1/2} \tag{5-8}$$

同时，它们的特征距离定义为：

$$\mathrm{dist_f}(\mathcal{P}_{l_1}, \mathcal{P}_{l_2}) = \left\| \underset{\mathcal{R}_m \cap \mathcal{P}_{l_1} \neq \varnothing}{\mathrm{avg}} \mathbf{d}_m - \underset{\mathcal{R}_m \cap \mathcal{P}_{l_2} \neq \varnothing}{\mathrm{avg}} \mathbf{d}_m \right\|_2 \tag{5-9}$$

将(5-8)式和(5-9)式组合，就得到总距离：

$$\mathrm{dist}(\mathcal{P}_{l_1}, \mathcal{P}_{l_2}) = \frac{\mathrm{dist_g}(\mathcal{P}_{l_1}, \mathcal{P}_{l_2})}{\max \mathrm{dist_g}(\mathcal{P})} + \frac{\mathrm{dist_f}(\mathcal{P}_{l_1}, \mathcal{P}_{l_2})}{\max \mathrm{dist_f}(\mathcal{P})} \tag{5-10}$$

这里，两种距离都进行了归一化，以使得它们的权重大致相同。最后，我们将所有图像上所有的总距离进行平均，从而在数据集层面上得到任何两个部件l_1和l_2之间的距离：

$$\mathrm{dist}(l_1, l_2) = \underset{\mathcal{P}_{l_1}, \mathcal{P}_{l_2} \neq \varnothing}{\mathrm{avg}} \mathrm{dist}(\mathcal{P}_{l_1}, \mathcal{P}_{l_2}) \tag{5-11}$$

我们定义一个需要学习的中层结构$\mathcal{L}_\mathrm{s}$，它包含一些基础部件：

$$\mathcal{L}_s = \{l_{s_1}, l_{s_2}, \ldots, l_{s_T}\} \tag{5-12}$$

其中$2 \leqslant T \leqslant L$，构建$\mathcal{L}_\mathrm{s}$的花费函数定义为：

$$\mathrm{cost}(\mathcal{L}_s) = \frac{\sum_{1 \leqslant i < j \leqslant T} \mathrm{dist}\left(l_{s_i}, l_{s_j}\right)}{\frac{1}{2} T\,(T-1)} \tag{5-13}$$

有了上述准备，**层次化结构学习**（Hierarchical Structure Learning，HSL）算法就很容易实现。

1. **初始化。** 原始的部件集合：$\mathcal{L} = \{1, 2, \ldots, L\}$；一个事先定义的学习参数$\mu$。
2. **学习。** 枚举$\mathcal{L}$的所有子集$\mathcal{L}_s$，计算其花费函数$c = \mathrm{cost}(\mathcal{L}_s)$。如果$c \leqslant \mu$，那么$\mathcal{L}_s$就被认为是一个新找到的中层结构。
3. **构建。** 将所有原始部件和找到的中层结构，组织为一个层次化的语义树。

记$\widetilde{\mathcal{L}} = \left\{1, 2, \ldots, L, L+1, \ldots, \widetilde{L}\right\}$为所有部件集合，包括原始存在的以及新找到的部件，这里有$\widetilde{L} \geqslant L$。

显然，随着学习参数μ的增加，学习到的中层概念也会相应增加，因此结构会变得更加复杂。我们在表5.3中罗列了一些较小的μ值，以及相应学习到的语义结构。可以看到，大部分学习到的结构都是可命名的，亦即，我们的算法一般能够找到一些确实具有语义的中层结构。在$\mu = 0.5$和$\mu = 1.0$时，我们还能够学习到层次化的结构，即某些中层结构包含在其他更高层的结果之中。

编号	μ	学习到的中层语义结构
#0	0.0	没有任何中层结构。
#1	0.1	**eyes** (*left*/*right eye*)、**legs** (*left*/*right leg*)、**wings** (*left*/*right wing*)。
#2	0.3	**eyes**、**legs**、**wings**、**neck** (*nape*/*throat*)。
#3	0.5	**eyes**、**legs**、**wings**、**neck**、**head** (*beak*/*crown*/*forehead*/**eyes**)、**body** (*back*/*belly*/*breast*/*tail*)。
#4	1.0	**eyes**、**legs**、**wings**、**neck**、**head**、**body**、(**wings/legs**)、(**body/wings/legs**)、**全部**。

表 5.3　学习参数μ和学习到的中层语义，其中粗体表示学习到的概念的命名。

HSL算法需要枚举所有$\mathcal{L}$的子集$\mathcal{L}_s$，并且花费部件个数平方的时间用于检查。这样，算法的时间复杂度不高于$O\left(L^2 \times 2^L\right)$：它随着部件个数$L$的增加呈超指数增长。幸运的是，在通常的视觉概念中，L通常都比较小（很少存在细粒度视觉概念会拥有超过20个部件）。我们的算法在$L = 20$时耗时大约100秒，而在这个例子（$L = 15$）里只需要不到3秒。考虑到所有的图像只需要处理一次，这样的复杂度是可以接受的。

5.3.5　几何池化策略

回到原图**I**。在对所有特征进行编码后，我们能够得到一个特征集合$\mathcal{W}$；同时，层次化结构学习还会产生$\widetilde{L}$个区域$\left\{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_{\widetilde{L}}\right\}$。

此时，朴素的池化方法直接寻找每个区域对应的特征：

$$\mathcal{W}_l = \{(\mathbf{w}_m, \mathcal{R}_m) \mid \mathcal{R}_m \cap \mathcal{P}_l \neq \varnothing\} \tag{5-14}$$

接着进行池化（如最大池化）：

$$\mathbf{F}_l^{\mathrm{W}} = \max_{(\mathbf{w}_m, \mathcal{R}_m) \in \mathcal{W}_l} \mathbf{w}_m \tag{5-15}$$

虽然朴素池化方法实现简单，但是它通常会忽略丰富的几何信息，而这些几何信息往往对于细粒度分类非常关键。图5.8展示了一些有助于鸟类分类的几何信息，如*crown*的形状和*tail*的长度。因此，我们采用第4章叙述的局部特征加强算法，以局部视觉短语的形式考虑特征之间的关系，以求设计一种有效的算法，加强对于特征之间几何关系的描述。

简单地说，GPP算法在每个特征组$\mathcal{W}_l$上建立一系列的局部特征组合，称为视觉短语。对于每个$\mathcal{W}_l$中的单词$(\mathbf{w}_m, \mathcal{R}_m)$，我们寻找它在$\mathcal{W}_l$里的$K$个最近邻（以图

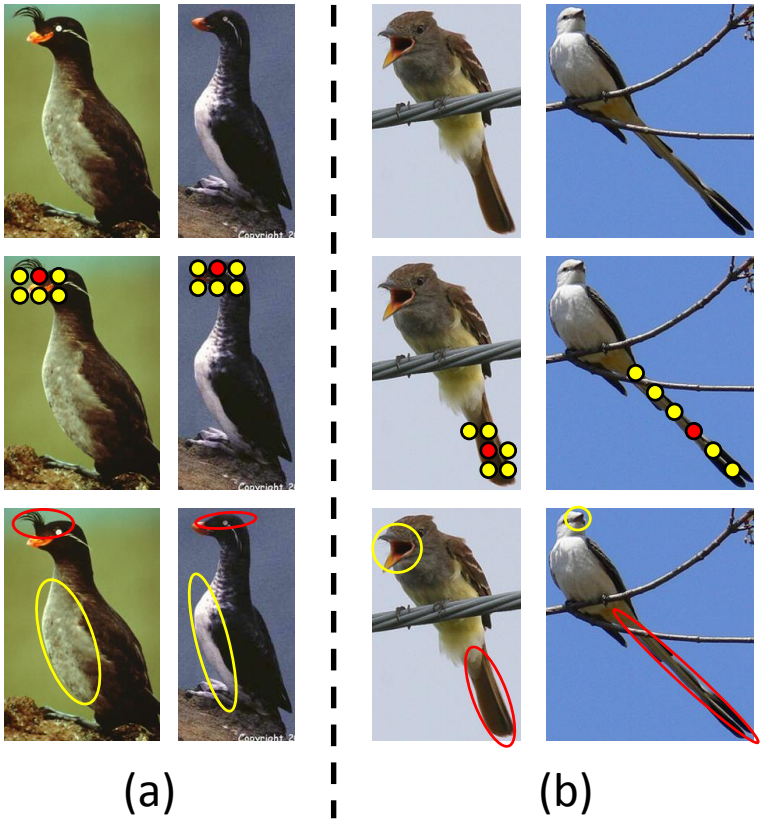


图 5.8　GPP算法的演示实例。上排：两组鸟类图像，其中最显著的区别分别体现在*crown*的形状和*tail*的长度。中排：提取的一些几何短语，其中红色圆圈表示中心单词，而黄色圆圈表示边缘单词。由于部件形状的不规则形，在这些部件中抽取的几何短语也呈现不规则的分布。下排：判别性显著增加的区域（直观判断的*crown*和*tail*都能够被检测到）。

像平面上的欧氏距离为度量），建立视觉短语：

$$\mathcal{G}_{l,m} = \{(\mathbf{w}_{l,m,0}, \mathbf{l}_{l,m,0}), \ldots, (\mathbf{w}_{l,m,K}, \mathbf{l}_{l,m,K})\} \tag{5-16}$$

$\mathcal{G}_{l,m}$是对应于$\mathcal{W}_l$里第m个单词的几何视觉短语（Geometric Visual Phrase，GVP）。其中，$\mathbf{w}_{l,m,0} = \mathbf{w}_m$是短语的中心单词，其余则是边缘单词，$K$是$\mathcal{G}_{l,m}$的大小。图5.8展示了一些视觉短语的例子。

几何短语池化（Geometric Phrase Pooling，GPP）算法对于短语$\mathcal{G}_{l,m}$计算如下向量值：

$$\mathbf{p}_{l,m} = \mathbf{w}_{l,m,0} + \max_{1 \leqslant k \leqslant K} \mathbf{w}_{l,m,k} \tag{5-17}$$

并且通过最大池化将所有短语组合起来：

$$\mathbf{f}_l^{\mathrm{P}} = \max_{(\mathbf{w}_m, \mathcal{R}_m) \in \mathcal{W}_l} \mathbf{p}_{l,m} \tag{5-18}$$

我们用一个定性的实验来展示GPP在细粒度分类问题上的效果。在图5.8展示的所有对应部件中，我们计算对应视觉单词和视觉短语之间的距离$\boldsymbol{\phi}_l$，并且将$\boldsymbol{\phi}_l$（由两张属于不同类的图像计算而得）视为模型的判别能力的度量：

$$\boldsymbol{\phi}_l^{\mathrm{W}} = \left\|\mathbf{f}_l^{\mathrm{W}}(\mathbf{I}_1) - \mathbf{f}_l^{\mathrm{W}}(\mathbf{I}_2)\right\|_2^2 \tag{5-19}$$

$$\boldsymbol{\phi}_l^{\mathrm{P}} = \left\|\mathbf{f}_l^{\mathrm{P}}(\mathbf{I}_1) - \mathbf{f}_l^{\mathrm{P}}(\mathbf{I}_2)\right\|_2^2 \tag{5-20}$$

我们计算$\boldsymbol{\phi}_l^{\star}$的增量，即$\boldsymbol{\phi}_l^{\mathrm{P}} - \boldsymbol{\phi}_l^{\mathrm{W}}$，显然，$\boldsymbol{\phi}_l^{\star}$越大，说明模型的判别能力越强。并且在图5.8中圈出增量最大的部分。可以看出，GPP算法确实对于那些判别性较强的区域比较敏感（相应部位的判别能力都得到了提升）。

5.3.6　实验部分

本节主要展示我们的算法在Caltech-UCSD **Birds-200**-2011数据集[34]上的分类结果。为了与以往的算法公平比较，我们采用一些通用的设置。

- **局部描述子。** 一张图像首先被重置大小，在保证其长宽比不变的情况下，将其长边重置为300个像素。我们使用VLFeat程序库[174]抽取密集的OpponentSIFT特征[70]。密集采样中，相邻特征的跨度为6像素，而特征的滑动窗口边长为12像素。
- **码本训练。** 我们用K-聚类算法训练大小为2048的码本。用于训练码本的描述子个数一般不超过2百万个。
- **特征编码。** 我们使用LLC[9]算法进行特征编码，基向量个数设定为$K = 5$。GPP算法的相应参数将在后面进行讨论。
- **特征组合。** 我们在每个检测到的基础部件和中层部件内分别执行最大池化，并且分别进行归一化操作[94]。
- **分类。** 我们使用LibLINEAR[99]，一个可扩展的支持向量机（SVM）模型，进行训练和测试。SVM的松弛参数始终设置为10。
- **精度计算。** 我们在每类随机选取5、10、20和30张训练图像，并且将其余图像用于测试。我们重复10次随机的训练/测试数据分割，并且报告10次的平均准确率。此外，数据集还提供了一个固定的训练和测试分割，我们也将在这个固定设定上进行实验。

5.3.6.1　手工和自动标注

在我们的算法中，手工弱标注信息是很重要的一环。然而，在实际应用中，期盼所有图像都具有弱标注信息并不现实。因此，许多相关工作[34][98][205][206]都

算法	详细描述	分类精度
基准算法	不使用部件的视觉词袋模型	13.64%
VOC07模型	**VOC2007**预训练的模型（9个部件）	9.43%
VOC11模型	**VOC2011**预训练的模型（9个部件）	11.09%
3个部件	保留原始15个部件中的3个	21.37%
6个部件	保留原始15个部件中的6个	23.91%

表 5.4　不同模型的分类准确率（%，每类的训练样本数为5）。自动检测的部件模型产生的分类结果甚至比基准结果还要差；同时，保留少数几个主要的标注就能够产生很好的分类结果。

训练样本数	5	10	20	30
基准方法	13.64	20.25	28.36	33.63
使用前景推断	19.25	27.66	37.08	43.06
使用部件切分	**28.55**	**40.46**	**52.52**	**58.09**

表 5.5　使用不同模型产生的分类准确率（%）。当我们不使用部件切分（前两个模型）时，我们使用一个3层的SPM模型。

在讨论使用人工标注的合理性。但在这里，我们简要论述人工弱标注信息对于细粒度分类问题的必要性。

作为与手工标注的对比，我们使用可形变的部件模型（Deformable Part Model，DPM）进行自动的部件检测。我们使用在**PascalVOC2007**和**PascalVOC2010**数据集上预训练的鸟类模型（包含总共9个部件）对**Bird-200**数据集进行检测。同时，为了弱化人工的标注，我们提供两个标注子集，分别保留3个和6个重要的标注部件（**Bird-200**数据集总共有15个部件）。

表5.4展示了两种方式分类结果的对比。我们可以看到，以利用DPM模型[41]获得的中心点作为标注点，产生的分类结果非常差，甚至比不上基准结果。另一方面，只需要保留3个或者6个主要的标注部件，就可以对分类结果产生巨大的提升。于是，我们得出结论：除非能够设计出更加强力的检测算法，否则人工弱标注信息对于细粒度分类问题是十分必要的①。

5.3.6.2　模型和参数

首先，我们测试前景推断和部件分割的效果。表5.5的结果表明，这两种方法都能够非常显著地提升分类准确率。

① 此论述基于本文发表时的研究进展，现时已经不再成立。请参见第5.3.7节的附注。

训练样本数	5	10	20	30
结构#0	28.55	40.46	52.52	58.09
结构#1	29.29	41.62	53.36	59.24
结构#2	29.75	42.03	53.55	59.32
结构#3	**30.33**	**42.66**	**53.94**	**59.86**
结构#4	27.38	38.64	50.22	56.11

表 5.6　中层结构的建立有助于提升分类准确率（%）。不同的中层结构的内容请参看表5.3。

训练样本数	5	10	20	30
不使用视觉短语	30.33	42.66	53.94	59.86
GPP(5,5)	31.69	43.80	55.26	60.80
GPP(5,10)	32.23	45.10	56.11	61.93
GPP(5,20)	34.13	47.29	58.60	64.01
GPP(5,40)	**36.09**	**48.87**	**60.56**	**65.62**

表 5.7　抽取空间几何短语结构有助于提升分类准确率（%）。括号内的数字表示对于中心单词和边缘单词分别使用的编码基个数（见第4.3.3节）。

接着，我们测试层次化结构学习（Hierarchical Structure Learning，HSL）算法。我们使用表5.3中陈列的中层结构，分类结果如表5.6所示。可以看出，使用中层结构相当于提供额外的池化箱，一般能够有效地提高分类准确率。例外出现在最后一种情况（结构#4）：由于结构过于复杂，出现了过拟合现象，反而导致分类精度的下降。在接下来的实验中，我们将始终使用结构#3。

我们还测试了几何短语池化（Geometric Phrase Pooling，GPP）算法的效果。结果如表5.7所示。可以看出，GPP作为一种强化的空间编码方法，确实有效地提升了分类准确率。我们选择其中最优的实验参数，即中心单词和边缘单词的编码基个数分别为5和40。

将上述几个模块结合，就得到了**层次化部件匹配**（Hierarchical Part Matching，HPM）算法，显著地提升了细粒度分类的效果。我们指出，HPM的成功主要来源于它对物体部件的准确捕捉。我们的实验同时还表明，强大的图像局部编码方式（GPP）可以和准确的图像空间分割有效地结合，同时对分类提供帮助。

5.3.6.3　与之前方法的对比

最后，我们将HPM算法与之前发表的工作进行对比。我们继承了前面章节里学习到的最好参数，即：使用前景推测和部件切分，使用层次化结构#3并且使

训练样本数	5	10	20	30
Wah[34]	10.05	-	-	-
Wang[9]	13.64	20.25	28.36	33.63
Xie[71]	15.34	22.91	31.01	36.17
我们的方法（均值）	**36.09**	**48.87**	**60.56**	**65.62**
我们的方法（标准差）	±0.31	±0.60	±0.50	±0.46

表 5.8　**Bird-200**数据集上，使用随机的训练测试分割的分类结果（%）。[9]是我们的基准系统，使用简单的LLC编码配合SPM。

Wah[34]	Zhang[98]	Wang[9]	Berg[89]	我们的方法
17.31	24.21	33.91	73.30	66.35

表 5.9　**Bird-200**数据集上使用固定的训练测试分割（每类大约有30张训练图像）的分类结果（%）。

用5和40作为GPP算法对应的基向量数量。表5.8和表5.9分别展示了算法在随机和固定的训练测试数据划分上的结果。我们可以看到，HPM算法显著地提升了现有的分类准确率。新方法[89]产生的结果与我们的方法相当：此方法也充分利用了物体部件的检测，以对齐不同图像的描述子。

5.3.7　结论

本节提供了一种新颖的，专门用于细粒度分类问题的结构，称为层次化部件匹配（Hierarchical Part Matching，HPM）。HPM包含若干新模块，包括前景推断和部件切分、层次化结构学习以及几何短语池化，用于捕捉更加丰富的空间信息，以提升分类效果。实验结果表明，HPM算法确实能够大幅提升细粒度分类的准确率。这启发我们投入更多的精力，研究如何以较小的代价，学习或者检测细粒度物体的部件信息。

注：本节相关工作于*2013*年底发表*[192]*。在此之后，许多无监督的部件检测算法（如*[89][90][91]*）都展现出了很好的效果。如今，在深度学习的帮助下，在细粒度物体识别数据集上的自动检测已经达到了非常高的水准。基于有效的检测算法，在不提供任何人工标注信息（包括包围框和部件标注点）的情况下，在细粒度分类数据集上的识别精度已经和*HPM*算法相当。本文确实在一定程度上，预测了一段时间内解决细粒度分类问题的发展趋势；时至今日，针对细粒度任务的部件检测依然是计算机视觉的热点问题，它还将吸引许多研究者的兴趣。当然，本节论述的一部分结论（如人工标注是不可或缺的）在今日已经不再成立。

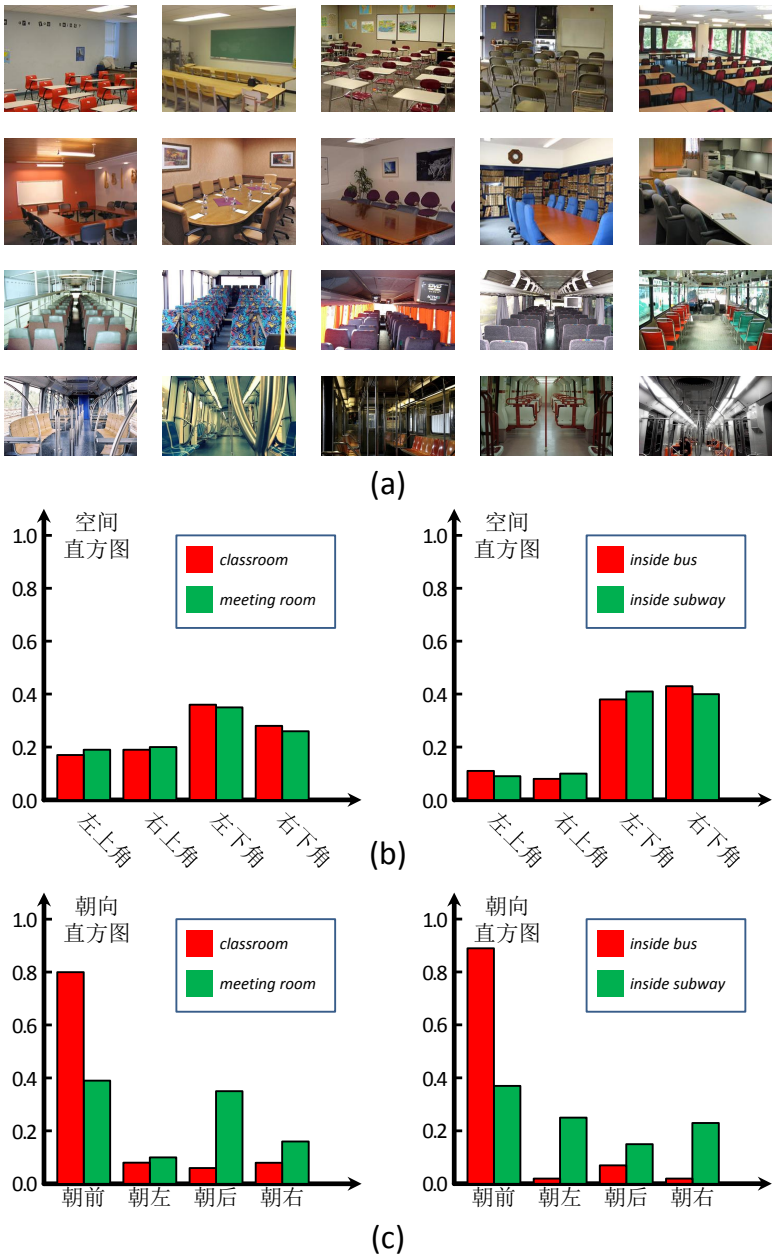


图 5.9　(a)MIT **Indoor-67**数据集[189]上4类场景的代表图像： 教室、会议室、地铁车厢和公交车厢（按照从上到下的顺序）。在每一对视觉场景中，使用朝向信息提取的特征都比使用空间信息提取的特征更具有描述力。 (b)(c)在每一对场景中，椅子的空间分布都非常接近，但是朝向分布却相差许多。我们对每一类中若干图像内的每一张椅子进行了手工计数，并且将其空间和朝向信息归纳为图中所示的四个类别之一。

5.4　场景分类：朝向金字塔匹配

5.4.1　问题综述

场景分类是计算机视觉中的一个基本问题。在此问题上，已经有一些传统方

法能够取得很好的结果，如视觉词袋模型[2]、物体银行（Object Bank，OB）模型[199]、部件包（Bag-of-Parts，BoP）模型[185]，等等。空间金字塔匹配（Spatial Pyramid Matching，SPM）[10]也能够与视觉词袋模型有效地结合。然而，这些方法在一些对精确性要求较高的分类问题上往往无法取得良好的精度，特别是在包含一些较为相似的场景的任务中。

图5.9(a)中的图像都来源于MIT **Indoor-67**数据集[189]。上部两行展示了两个相似类（教室和会议室）的图像样例。我们可以观察到，这两个类的图像包含的视觉概念几乎相同，都有一些椅子、桌子、墙壁以及日光灯，这些物体在视觉上非常类似，导致这两类图像的特征表示也非常相似，难以被区分开。然而我们可以注意到，这两类图像的主要区别在于物体（如椅子）的朝向：教室里的椅子通常朝向同一个方向，这样所有学生都可以面向讲台或者黑板；而会议室里的椅子则往往朝向不同的方向，以方便参会人的互相讨论。类似的情形还发生在其他类别上，例如在图的下半部展示的地铁车厢和公交车厢的图像对比。

图5.9(b)(c)展示了一些定量分析的结果，即教室、会议室、地铁车厢和公交车厢四个类中椅子的分布情况。我们随机选取每类的100张图像，并且手工对出现在不同位置、朝向不同方向的椅子进行了计数。从统计结果上看，在教室和会议室、地铁车厢和公交车厢这两对场景概念中，椅子的朝向分布比空间分布更具有区分能力——这也构成了这些场景对的主要区分特征。

上述观察说明，对于物体的（三维）朝向进行建模，将有助于场景分类任务的进行。为此，我们利用一个数据驱动的模型[207]以估计场景中物体的三维朝向。我们提出**朝向金字塔匹配**（Orientational Pyramid Matching，OPM）模型，以增强场景图像特征编码中的朝向信息。与我们熟悉的SPM算法一样，OPM层次化地将图像中的特征集合进行分组，只不过此处用于分组的标准不再是空间位置，而是朝向向量。实验结果表明，OPM算法产生的特征与SPM算法产生的特征描述能力相当，且具有很强的互补性。结合两种特征，我们就可以在若干极具挑战性的数据集上，提升现有的场景分类精度。

下面我们分两小节介绍朝向金字塔匹配算法。

5.4.2　朝向金字塔匹配

给定一个密集采样的特征集合$\mathcal{W}$，我们的目标是将它们组合成为一个特征向量。SPM算法利用空间信息将特征分为S组，并且在每个组内实施池化算法，以获得S个独立的特征向量。详细的SPM算法介绍见第5.2.2节。与SPM不同的是，OPM算法计算每个特征对应的朝向向量，并且利用朝向信息进行特征的分组。由于3D朝向可以表示为一个单位三维向量，我们将其记为$\mathbf{o} = (\theta, \varphi)^{\top}$，其中$\theta$和$\varphi$分

别表示*方位角*（azimuth angle）和*极角*（polar angle）。记带有朝向信息的特征集合为：$\mathcal{W} = \{(\mathbf{w}_1, \mathbf{o}_1), (\mathbf{w}_2, \mathbf{o}_2), \ldots, (\mathbf{w}_M, \mathbf{o}_M)\}$。

这样，OPM算法就依据朝向将$\mathcal{W}$切分为若干子集$\{\mathcal{W}_t\}$，$t = 1, 2, \ldots, T_{\mathrm{O}}$，其中每个子集都包含一些朝向相似的特征（SPM切分的每个子集包含一些位置相似的特征）。我们使用一种非常简单的方法进行朝向空间切分。根据球面坐标的定义，朝向空间可以表示为$\mathcal{U} = \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]^2$，将其切割为一些规则的矩形。令$L_{\mathrm{A}}$和$L_{\mathrm{P}}$分别表示沿方位角和极角方向的切割份数，那么第$l$层的每个池化箱的大小就是：$\frac{\pi}{2^{\min\{l, L_{\mathrm{A}}\}}} \times \frac{\pi}{2^{\min\{l, L_{\mathrm{P}}\}}}$。也就是说，在第$l$层上，总共有$2^{\min\{l, L_{\mathrm{A}}\}} \times 2^{\min\{l, L_{\mathrm{P}}\}}$个池化箱。这样的一个$L$层模型就形成一个**朝向金字塔**（orientational pyramid）。

假设朝向金字塔划分成的子集为：$\mathcal{W}_1, \mathcal{W}_2, \ldots, \mathcal{W}_{T_{\mathrm{O}}}$。每个子集$\mathcal{W}_t$包含$M_t$个局部特征：$\{\mathbf{w}_{t,1}, \mathbf{w}_{t,2}, \ldots, \mathbf{w}_{t,M_t}\}$。我们将这$M_t$个特征总结为一个向量$\mathbf{f}_t$，用于描述$\mathcal{W}_t$。同前面一样，我们仍然对LLC编码使用最大池化：$\mathbf{f}_t^{\mathrm{LLC}} = \max_{m=1}^{M_t} \mathbf{w}_{t,m}$；而对Fisher向量编码使用求和（平均）池化：$\mathbf{f}_t^{\mathrm{Fisher}} = \sum_{m=1}^{M_t} \mathbf{w}_{t,m}$. 最终，将这些独立的区域向量拼接起来，就得到全图的向量表示。

图5.10提供了SPM和OPM算法的一个简单直观的对比。两种算法的计算方式非常类似，唯一的区别就是它们对特征分组的不同准则：OPM使用朝向信息，而SPM使用位置信息。两种算法也具有一些类似的性质，例如在$L_{\mathrm{X}} = L_{\mathrm{A}}$以及$L_{\mathrm{Y}} = L_{\mathrm{P}}$时，编码向量的长度相等。这里，$L_{\mathrm{X}}$和$L_{\mathrm{Y}}$分别表示沿$x$和$y$方向上的切割份数。 SPM和OPM的时间复杂度是非常类似的，因为它们都只需要对所有的特征检查一遍。

应用在SPM上的大部分方法都可以应用在OPM上。例如，我们可以基于朝向金字塔，学习相应的朝向感受野（receptive fields）[87] 或者姿态相关的池化核（pose pooling kernels）[98]。本节将只研究OPM与LLC[9]和Fisher向量[7]编码的配合情况。

5.4.3　计算3D朝向

我们应用一个数据驱动（data-driven）的算法[207]以判断图像区块的3D朝向。简单地说，我们的方法以近邻搜索为基础，参考每个测试区块在训练集中最近邻的朝向，来估计其朝向。

我们也使用同一论文[207]中提出的**Bristol**数据集训练朝向估计模型。该数据集中的每一张图像都附带一些人工标定的区域，在各区域内提供3D朝向向量（或声明该区域不是平面区域）。我们将每个平面区域的3D向量$(x, y, z)^{\mathsf{T}}$（满足$x^2 + y^2 + z^2 = 1$）转化为方位角和极角的形式，由于z向量恒非负（所有向

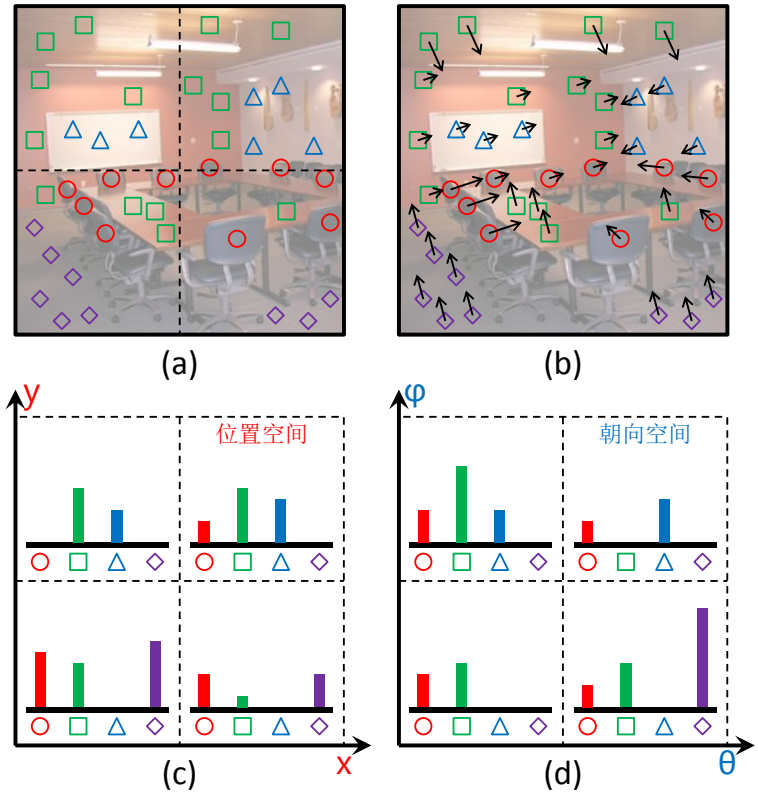


图 5.10　在第1层上，SPM和OPM的一个直观对比。 (a)局部特征根据其表象特征，被量化为4种类型； (b)赋予这些局部特征不同的位置和3D朝向向量； (c)(d)根据位置和朝向信息产生的SPM和OPM池化结果。由于每种特征的空间和朝向分布不同，SPM特征和OPM特征产生了一定的区别，尤其是紫色菱形（表示地板特征）。

量都默认朝向纸外方向），这些单位3D向量都落在一个半球面上，即同时满足$\theta = \arctan\left(\frac{z}{x}\right)$和$\varphi = \arcsin(y)$。

随后，我们以密集采样的方式在训练图像中提取区块$\{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_M\}$。每个区块都被赋予对应的平面性质：完全落在平面区域内，完全落在非平面区域内，或者位于某个边界上，将这三类分别记为C_1、C_2和C_3。对于每个完全平面区块，我们还能够得到它对应的朝向向量。这样，每个区块$\mathbf{P}_m$都有一个SIFT特征$\mathbf{d}_m$、平面性标定$c_m \in \{C_1, C_2, C_3\}$、以及朝向向量$(\theta_m, \varphi_m)$。我们收集了100000个区块用于训练，包括50000个完全平面区块，30000个完全非平面区块及20000个边界区块。

在一个未知朝向性质的测试区块上，我们首先提取它的SIFT特征$\mathbf{d}$。随后，我们寻找它在训练集上的K个最近邻特征（以SIFT特征空间中的欧氏距离度量），并且用它们的平面性质来估计测试区块的性质。如果在K个最近邻区块中，有至少$\tau = \frac{K}{2}$个区块是完全平面区块，我们就认为测试区块也是平面的；此时，我们对所有最近邻平面区块的朝向向量取平均值，作为测试区块的朝向向量。如果没有

K值	c-score	r-score
0	0.5000	0.5676
1	0.7694	0.6458
5	0.8653	0.6882
10	0.8872	0.6946
50	0.8896	0.6980
100	**0.8902**	**0.6990**
500	0.8899	0.6985
1000	0.8896	0.6968

表 5.10　平面性分类精度（c-score）和朝向预测精度（r-score）与最近邻个数K的关系，$K = 0$表示随机猜测。

一半的最近邻区块满足平面性，那么我们就认为该测试区块是非平面的，这种区块在图像编码过程中将被直接丢弃（不参与池化）。

3D朝向的测试由两种标准组成：平面性分类精度，和朝向预测精度。平面性分类精度，或称**c-score**，是指平面性判断（测试区块是否为完全平面区块）的二分类准确率；而朝向预测精度，或称**r-score**，是指对平面区块朝向向量预测的回归准确率（内积在$[-1, 1]$区间上的规范化）。表5.10报告了平均的分类和预测精度与近邻参数K的关系。我们选取在此测试中表现最好的$K = 100$（及其产生的结果），用于后续的图像分类应用。

值得注意的是，**Bristol**数据集仅仅包含室外场景，而我们在实际分类问题中常常需要处理室内场景图片。使用室外场景的朝向标注来预测室内场景的朝向信息，这样的实验设置或许存在疑问，然而实验结果表明，在**Bristol**数据集上学习到的模型能够有效地估计室内场景的朝向信息。当然，我们也期望能够得到更多、更丰富的朝向标注数据，以训练更加鲁棒的朝向预测模型。

5.4.4　实验部分

在这一部分，我们首先测试我们提出的方法（OPM）在一个室内场景分类数据集上的结果；随后，我们将实验扩展到一般化的室内外场景分类数据集，以及一般物体识别的数据集，并且做出相应的讨论。

5.4.4.1　数据集和实现细节

我们使用两个通用的场景分类数据集。 MIT **Indoor-67**数据集[189]是当前最流行的室内场景分类数据集，包含67个类，15620张图像。图5.9展示了该数据集的样例图像。 **SUN-397**数据集[31]则是当前最大的场景分类数据集，包含397个室内

室外场景类和超过10万张图像。我们在每一类上随机选取一定数量的训练图像和一定数量的测试图像，并且报告按类平均的分类准确率。在**Indoor-67**数据集上，每一类的训练和测试图像数量分别为80和20；在**SUN-397**数据集上，每一类的训练和测试图像数量分别为50和50。我们将随机进行10次训练测试图像划分，并且报告10次的平均分类准确率。

实验的基础设定遵循近期发表的视觉词袋模型[9]（LLC编码）和[19]（Fisher向量编码）。一张图像首先被重置大小，在保证其长宽比不变的情况下，将其长边重置为600个像素。我们用VLFeat[174]，一个通用的计算特征代码库，提取密集的RootSIFT特征[118]。我们抽取两个SIFT特征集合，密集采样的空间跨度分别为8和16，窗口大小与空间跨度保持相同。为了进行Fisher向量编码，128维的SIFT向量被PCA降维至64维。对于LLC编码，我们用K-Means训练一个具有8192个单词的码本；而对于Fisher向量编码，我们训练一个具有256个分量的GMM。用于聚类的SIFT特征个数大约是5百万。对于LLC编码，我们附加使用第4章介绍的GPP模型，用以加强局部的特征编码。我们同时采用SPM和OPM模型用以进行位置空间和朝向空间上的特征组合。对于LLC编码和Fisher向量编码，我们分别采用3层和2层的金字塔模型。所有图像的向量表示最后被送入LibLINEAR[99]，一个通用的线性SVM模型，以进行训练和测试。在SVM中，松弛参数始终设置为10。

5.4.4.2　模型和参数

此处，我们分析不同的L_{A}和L_{P}（朝向金字塔的“宽度”）对于MIT **Indoor-67**数据集分类精度的影响。我们对比三种模型：SPM模型、OPM模型、以及将SPM和OPM产生的向量拼接起来的模型（记为OPM+SPM）。注意，我们始终保持$L_{\mathrm{X}} = L_{\mathrm{A}}$和$L_{\mathrm{Y}} = L_{\mathrm{P}}$，以确保SPM和OPM产生的特征维度一致。这样，我们就相当于使用同样大的力度来表达图像的位置和朝向信息。

不同模型产生的结果罗列在表5.11和表5.12中。我们可以观察到，分类精度随着位置和朝向池化向量的增加而增长。为了避免特征维度太高，我们在LLC编码时选择$L_{\mathrm{X}} = L_{\mathrm{Y}} = L_{\mathrm{A}} = L_{\mathrm{P}} = 3$，而在Fisher向量编码时选择$L_{\mathrm{X}} = L_{\mathrm{Y}} = L_{\mathrm{A}} = L_{\mathrm{P}} = 2$。我们还能够观察到，单独使用OPM产生的分类结果比单独使用SPM产生的分类结果略差一些，这表明位置信息比朝向信息更加重要。将SPM和OPM产生的向量拼接起来以后，我们能够得到比单独模型更高的分类精度，说明SPM和OPM提供了一些互补的信息，有利于图像分类。

$(L_X, L_Y) = (L_A, L_P)$	SPM	OPM	OPM+SPM
(1, 1)	41.93	41.93	41.93
(1, 2)	49.10	43.57	52.35
(1, 3)	54.10	43.75	56.14
(2, 1)	48.78	46.31	52.98
(2, 2)	53.55	46.47	56.09
(2, 3)	55.42	46.48	57.30
(3, 1)	53.61	48.64	57.20
(3, 2)	55.09	48.79	58.19
(3, 3)	57.83	48.83	**59.57**

表 5.11　不同的模型配合LLC编码在MIT **Indoor-67**数据集上的分类精度（%）。

$(L_X, L_Y) = (L_A, L_P)$	SPM	OPM	OPM+SPM
(1, 1)	46.63	46.63	46.63
(1, 2)	57.21	48.25	59.14
(2, 1)	56.47	50.99	59.55
(2, 2)	61.22	51.45	**63.48**

表 5.12　不同的模型配合Fisher向量编码在MIT **Indoor-67**数据集上的分类精度（%）。

5.4.4.3　与之前工作的比较

我们在MIT **Indoor-67**数据集上比较我们的算法和其他算法的准确率。由于LLC编码[9]和Fisher向量编码[7]产生的分类准确率差别较大，我们将使用两者的结果分别列于表5.13和表5.14内。可以观察到，在两种情形下，我们的算法

算法	准确率
Quattoni[189]	26.0
Li[199]	37.6
Wang[9]	54.62
Xie[191]	57.83
Juneja[185] (BoP)	46.10
Juneja[185] (SPM+BoP)	56.66
Ours (OPM)	48.83
Ours (SPM+OPM)	**59.57**

表 5.13　使用LLC编码时，在MIT **Indoor-67**数据集上的分类结果（%）与之前方法的比较。基准算法为[191]。

算法	准确率
Perronnin[7]	61.22
Kobayashi[82]	58.91
Juneja[185] (SPM)	60.77
Juneja[185] (BoP)	46.10
Juneja[185] (SPM+BoP)	63.10
Ours (OPM)	51.45
Ours (SPM+OPM)	**63.48**

表 5.14　使用Fisher向量编码时，在MIT **Indoor-67**数据集上的分类结果（%）与之前方法的比较。基准算法为[7]。

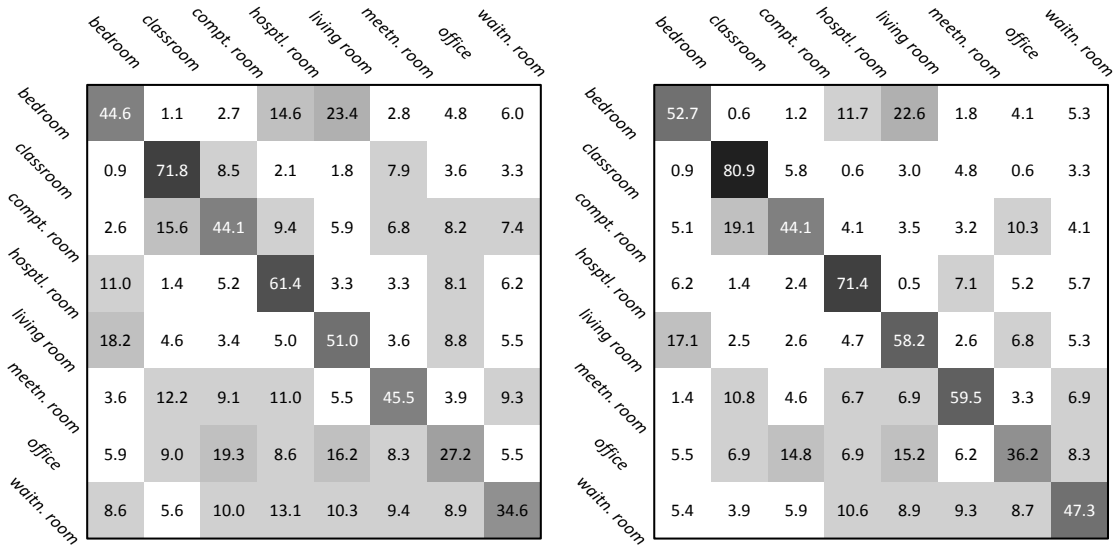


图 5.11　在使用LLC编码时，不使用OPM（左）和使用OPM（右）的混淆矩阵（%）对比。

都取得了很好的准确率。其中对于使用LLC编码的情形（表5.13），我们的方法在基准[191]上提升了接近2%。在表5.14中，近期发表的论文[185]使用BoP模型取得了与我们相当的准确率。注意到[185]将BoP特征加入SPM特征后，分类准确率从60.77%提升到了63.10%；而我们的方法将OPM特征加入SPM特征，则将分类准确率从61.22%提升到了63.48%。此外，BoP的单独分类准确率为46.10%，显著地低于OPM的分类准确率51.45%。这说明在同样为SPM特征提供互补信息的情况下，OPM特征的描述力和互补性都要高于BoP特征。

5.4.4.4　经验分析

我们为SPM特征和OPM特征的分类结果提供经验分析。我们首先观

类别A	类别B	OPM提升	混合特征提升
bookstore	*library*	+6.69%	+3.75%
auditorium	*concert hall*	+5.35%	+3.93%
computer room	*office*	+4.40%	+1.80%
jewellery shop	*lobby*	+2.90%	+1.06%
bakery	*buffet*	+2.88%	+2.18%

类别A	类别B	SPM提升	混合特征提升
gameroom	*garage*	+4.78%	+1.95%
restr. kitchen	*studiomusic*	+4.64%	+2.33%
library	*clothingstore*	+4.62%	+3.40%
children room	*kindergarden*	+4.47%	+1.91%
laboratorywet	*kindergarden*	+4.47%	+2.48%

表 5.15　一些场景概念对，各自使用OPM特征（上部）或SPM特征（下部）能够取得更好的分类结果。举例说明，在场景对*bookstore*和*library*上，使用OPM特征代替SPM特征能够产生6.69%的分类准确率提升，而使用混合特征代替SPM特征能够产生3.75%的准确率提升。

图 5.12　我们展示一些具有最大差别的场景对，以及在第1层上一些常见概念的位置和朝向分布。红色和绿色的直方图对应于相应颜色标题的图像上的物体分布。

算法	准确率
Xiao *et.al.* [31]	38.0
Sanchez *et.al.* [19]	43.2
Ours (SPM)	43.58
Ours (OPM)	34.61
Ours (SPM+OPM)	**45.91**

表 5.16　在**SUN-397**数据集上的分类结果（%）与之前方法的比较。

察MIT**Indoor-67**数据集中，SPM模型不能很好地区分的8个类：卧室、教室、机房、病房、起居室、会议室、办公室和等候室。图5.11的左右部分分别展示了单独使用SPM模型和使用两种模型混合，在这8个类上的分类混淆矩阵。可以观察到，在加入了OPM特征后，大部分混淆矩阵对角线外的元素都变小了，例如机房到医院的混淆值从9.4%减少到了4.1%；此外，大部分混淆矩阵对角线上的元素都变大了，例如会议室的分类准确率从45.5%提升到了59.5%。这说明OPM帮助区分了一些容易分错的概念，这些提升大多来源于对朝向信息的编码。

我们同时观察那些单独使用SPM和OPM特征难以区分的类。表5.15的上下部分分别展示了那些从OPM相对SPM提升最大的类，以及SPM相对OPM提升最大的类。为了观察空间和朝向信息在表示这些图像中的作用，我们进行了一些手工标注，将这些类中最常出现的物体量化为一个大小为8的码本，并且将这些物体的空间和朝向分布展示于图5.12中。可以观察到，在OPM分类效果更好的类中，朝向信息一般起到主导作用；而在SPM分类效果更好的类中，位置信息一般起到主导作用。这说明：OPM特征对于分类效果的提升确实来源于它对朝向信息的编码；同时，融合特征提升的精度确实得益于SPM特征和OPM特征的互补配合。

5.4.4.5　其他数据集

为了测试算法的可扩展性，我们将它应用于**SUN-397**数据集[31]。表5.16报告了使用Fisher向量编码的分类结果。我们再次观察到OPM特征对于场景分类精度带来的提升效果。同时，我们注意到**SUN-397**数据集同时包含室内和室外场景（室外图像通常包含更多的非平面区块）。能够在这样复杂的数据集上取得良好的效果，说明OPM特征也能够适用于室外场景识别任务。我们的准确率也超过了[31]：在这一工作中，多达11种特征被融合在一起，以提供图像的多方面描述。

我们同时测试算法在**Caltech101**[28]，一个一般化的物体识别数据集上的分类准确率。令人吃惊的是，在这样一个没有明显场景概念的数据集上，结合OPM特征同样能够带来准确率的提升（混合模型精度提升为1.02%）。这样的提升虽然比

算法	准确率
Chatfield *et.al.*[171]	77.78
Jia *et.al.*[87]	75.3
Ours (SPM)	80.73
Ours (OPM)	65.59
Ours (SPM+OPM)	**81.75**

表 5.17　在**Caltech101**数据集上的分类结果（%）与之前方法的比较。

较小，但是在我们报告的高基准（80.73%）上已经很不容易。我们认为，OPM产生的精度提升主要来源于它对一般物体所处场景的识别能力的提高。例如，一辆汽车通常出现在室外场景中，而一台电视则更有可能出现在起居室里。这就表明，OPM也能够为非场景分类任务提供辅助性特征。

5.4.5　结论

本节讨论了朝向特征在场景分类任务中的应用。我们提出了朝向金字塔匹配（Orientational Pyramid Matching，OPM）算法，以捕捉图像区块的3D朝向信息，用于特征组合；同时，我们将SPM和OPM结合在一起，并论证了它们对于图像表示的互补性。在MIT **Indoor-67**数据集和**SUN-397**数据集上，算法的分类准确率都达到了先进水平。此外，实验还表明，OPM特征也能够通过更加准确的场景建模，辅助一般物体的分类任务。

应当注意到，我们的算法在场景朝向分析上还存在一定的缺陷。当前采用的完全基于近邻查找和平均估计的算法过于简单，因而在很多情况下无法准确地预测朝向信息。在将来的研究中，我们应该探索更加有效的朝向预测算法，以使OPM算法获得更好的分类准确率。

5.5　本章小结

本章讨论了三个模型：推广的规范空间池化（Generalized Regular Spatial Pooling，GRSP）、层次化部件匹配（Hierarchical Part Matching，HPM）和朝向金字塔匹配（Orientational Pyramid Matching，OPM）。这些方法都是基于SPM模型，并且针对其朴素的特征分组进行一般性或者针对性的改进，来提升分类准确率。

我们的研究表明，优化特征组合方式是图像分类领域的一个重要课题。随着图像分类任务的精确化（细粒度分类、场景分类）和大规模化（海量图像分类数据集），特征组合算法对于分类效果的影响将愈加明显。

第 6 章　图像检索：特征索引和后处理

6.1　研究动机

大规模图像检索是一个具有巨大商业价值的实际问题。对于近似重复（near-duplicate）或者部分重复（partial-duplicate）的图像检索已经具有成熟的解决方案：视觉词袋模型。在这一模型中，大部分特征编码之前的步骤都与前述章节（用于图像分类的模型）非常类似，只不过为了提升大规模应用的效率（检索速度），采用了一种高效的数据结构（倒排表）用于存储和查找图像及相关特征。

本章研究在倒排表结构下，后处理模块对于检索精度的重要性。我们主要针对部分重复的检索问题，提出两种高效的方法，用于修正并改进图像检索质量，尤其是保证排名靠前的图像更具相关性。这两种方法分别称为**异质图传播**（Heterogeneous Graph Propagation，HGP）算法和**图像网络**（ImageWeb）算法。它们的共同特点是利用图像和特征之间的联系建立图结构，并且利用类似随机游走（random walk）的方式进行求解，以对检索结果重排序。其中，ImageWeb可以视为HGP的后续版本：它对问题的建模更简单，效果也更好。在一些大规模（百万级别）图像检索数据集上，我们的算法都显著地提升了基准方法的检索精度，从而验证了相应思想的有效性。

与本章相关的出版物为[110] [208]。

6.2　异质图传播算法

6.2.1　问题综述

尽管基于视觉词袋模型的图像检索流程简单高效而且可扩展，但是它们产生的结果通常不能达到很高的准确率（precision）或者召回率（recall）。其中，主要的原因包括局部特征（如SIFT）的描述力不足，以及在量化过程中产生的信息丢失。事实上，局部特征之间的匹配往往对于图像变换（如人工编辑、几何拉伸和变形等）比较敏感，从而导致许多情况下，相关的特征无法被量化到同一个单词，而不相关的特征却被错误地匹配在一起。上述现象导致的直接结果，就是在查询过程中，相关的图像被排序在不相关的图像之后，导致检索质量的显著下降。为了解决上述问题，后处理（post-processing）模块被广泛地应用于图像检索算法中，利用图像的额外信息（如特征的几何信息[75]、从初始排序靠前的图像中提取

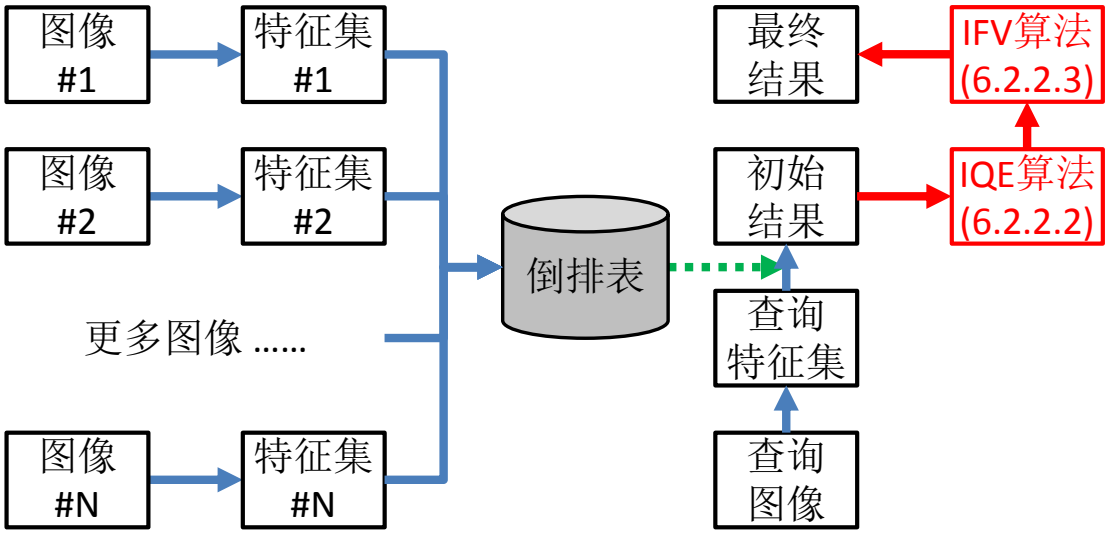


图 6.1　包含离线索引（左部）和在线查询（右部）的通用图像检索框架。我们提出的IQE和IFV算法用红色高亮表示。

的额外特征[23] 以及置信传播[24]），对检索结果进行重排序，从而达到提升检索质量的目的。

本文提出一种新的，基于图结构的视角，来审视图像检索的后处理模块。我们观察到，可以利用图像和特征之间天然存在的包含与被包含关系，通过置信传播的方式，来提升检索质量。为此，我们构建一个**异质**的图结构，显式地包含图像和特征两种节点。在这个图结构的基础上，我们提出两种算法（**增量查询扩展**（Incremental Query Expansion，IQE）和**图像特征投票**（Image-Feature Voting，IFV）），分别用于提升检索结果的召回率和准确率。图6.1展示了整个检索过程的流程图。我们的方法有两方面的优势：第一，IQE和IFV算法都是几何无关（geometry-free）的，即不需要利用任何特征的几何位置信息（如位置、大小、角度等），这样在实际的在线检索模块中，我们能够节省下可观的时间和空间开销；第二，我们的后处理方法与基准检索模块是相互独立的，可以与不同的基准算法配合（见6.2.3.2节）。同时，我们的方法还具有时间和空间的高效性与可移植性，因而能够很容易地应用在实际问题上。

6.2.2　异质图传播

本节从一个图结构的视角考察图像检索的后处理过程。我们建立了一个异质的图结构，并且在其上提出两种算法，分别提升检索的准确率和召回率。

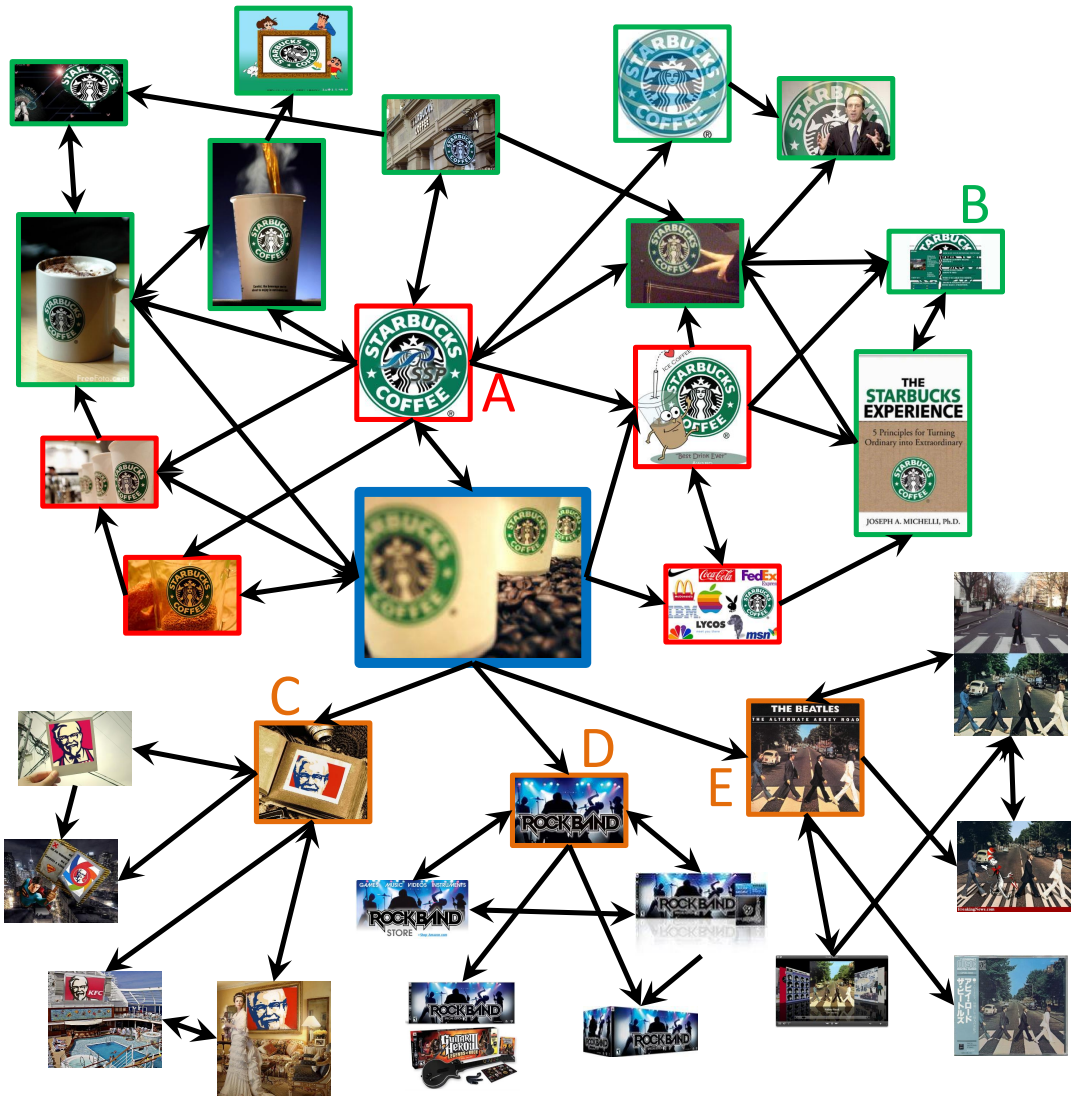


图 6.2　一个样例检索结果，包含了全部四类检索结果。

6.2.2.1　从同质到异质

首先，我们将检索结果可视化，展示为一个图结构（图6.2）。图中的每一个节点，都表示数据集中的一张图片：从图片**X**到**Y**有连边，当且仅当**Y**出现在以**X**为查询的初始检索结果的前10位。我们利用一个相对低质量（弯曲且模糊化）的*Starbucks Coffee*标志作为查询。图6.2展示了检索结果中可能出现的四类样本。

1. **真正（true-positive）样本**（红框，例如**A**）。这些相关样本与查询图像之间有许多匹配特征，因此自然地被排在检索结果靠前的位置。

2. **假负（false-negative）样本**（绿框，例如**B**）。这些相关样本与查询图像之间没有足够的匹配特征，因此被排在检索结果较为靠后的位置。我们希望将这些样本的排名提前，以提升检索结果的召回率。

3. **假正（false-positive）样本**（橙框，例如**C**、**D**和**E**）。这些不相关样本与查询图像之间有一定数量的匹配特征，因此被排在检索结果较为靠前的位置。我们希望将这些样本的排名推后，以提升检索结果的准确率。

4. **真负（true-negative）样本**（无框）。我们直接忽略此类样本，因为它们通常无法对检索结果产生影响。

显然，正是那些假正和假负样本的不正确排名，影响了检索结果的质量。一个理想的后处理模块，应该能够提升假负样本的排名（提高召回率），同时降低假正样本的排名（提高准确率）。初始检索结果一般通过排序候选图像与查询图像之间匹配特征的数量获得，但是并非所有特征都是同等重要的：如果一个特征**f**恰好位于图像中的感兴趣（包含相应语义）区域，那么它就应该被用于查找那些包含相同特征，却排序较低的假负样本；反之，它的权值就应当适当地减弱，以确保不会有一些假正样本因为包含**f**而排序较高。

鉴于特征在图像排序中的重要性，我们通过对每个特征建立对应的节点，显式地将它们加入图像重排序过程。这样，原先的同质（只含有表示图像的节点）图结构就转化为异质（同时包含表示图像和特征的节点）图结构。形式上，我们定义一个**异质图**为：$\mathbf{G} = \{\mathcal{I}, \mathcal{F}, \mathcal{E}, \mathbf{S}, \mathbf{W}\}$。其中，$\mathcal{I} = \{\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_N\}$和$\mathcal{F} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_M\}$表示图像和特征的集合，$\mathcal{E}$是无向边的集合。边只存在于图像节点和特征节点之间。图像$\mathbf{I}_n$和特征$\mathbf{f}_m$有边相连，当且仅当特征$\mathbf{f}_m$出现在图像$\mathbf{I}_n$里：即图像$\mathbf{I}_n$包含至少一个量化为特征$\mathbf{f}_m$的描述子。 **S**和**W**里的元素，分别表示图像的得分（score）和特征的权值（weight）。图6.4(c)展示了一个典型的异质图结构。下面，我们展示两个迭代算法，以计算异质图结构内的得分和权值。

6.2.2.2　提升召回率：增量查询扩展

我们从任意一个基于词袋模型和倒排表的初始检索结果开始。记查询图像为$\mathbf{I}_{q,0}$，而$\mathbf{f}_m \in \mathbf{I}_n$（或者等价的$\mathbf{I}_n \ni \mathbf{f}_m$）表示特征$\mathbf{f}_m$出现在图像$\mathbf{I}_n$中，亦即图像$\mathbf{I}_n$里至少有一个描述子被量化为$\mathbf{f}_m$。初始检索将所有出现在图像$\mathbf{I}_{q,0}$中的特征的权值设为1：

$$w(\mathbf{f}_m) = \mathbb{I}\left(\mathbf{f}_m \in \mathbf{I}_{q,0}\right) \tag{6-1}$$

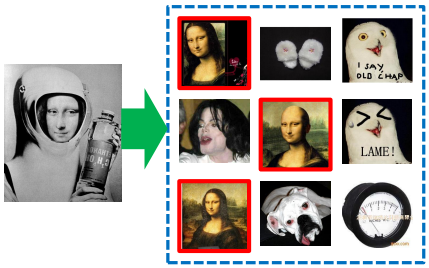
然后用累积的方式计算每张图像的分数：

$$s(\mathbf{I}_n) = \sum_{m, \mathbf{f}_m \in \mathbf{I}_n} w(\mathbf{f}_m) \tag{6-2}$$

(a)

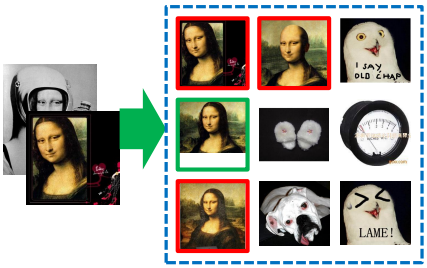
召回率位置	10	100	1000
扩展之前	0.05	0.28	0.62
扩展之后	0.12	0.85	0.93

(b)



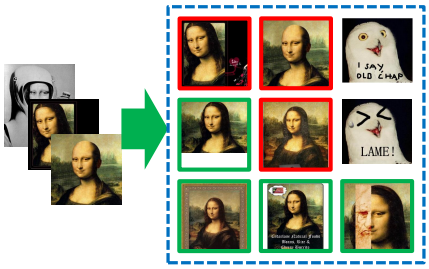
图像 所包含的特征 被用于扩展搜索

(c)



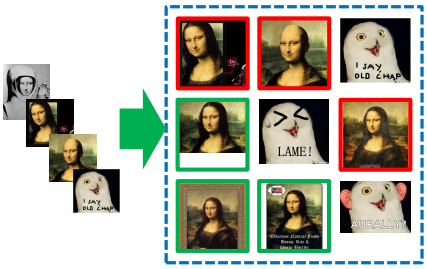
图像 所包含的特征 被用于扩展搜索

(d)



图像 所包含的特征 被用于扩展搜索

(e)



图像 所包含的特征 被用于扩展搜索

(f)

图 6.3　一个真实的增量查询扩展（IQE）过程。 (a)查询图像，其中只有一小部分区域（脸部）没有经过修改，因此能够找到的有效特征非常有限。 (b)在执行IQE算法前后，前10、前100和前1000位的召回率对比。 (c)初始检索结果：由于查询图像质量较差，因此初始结果中的正例（红框）非常少。此时，排名第一的检索结果（图像）被用于查询扩展。 (d)(e)(f)第1、第2和第3次查询扩展后的检索结果。其中，第3次用于查询扩展的图像是一个负例，因此引入了不少同为负例的图像。然而，我们通过3次查询扩展，找到了比原先更多的正例（红框）。在IQE算法中引入的负例，将通过后续的算法有效地筛除。

对初始分数排序，就得到了初步（查询扩展前）的检索结果：

$$\mathcal{I}_0 = \{\mathbf{I}_{0,1}, \mathbf{I}_{0,2}, \ldots, \mathbf{I}_{0,N}\} \tag{6-3}$$

然而，由于局部特征的描述能力有限，因此上述简单的方法在许多情况下无

法准确地区分假正和假负样本。直观的观察表明，在图6.2中，我们往往能够在查询图像和假负例之间，找到一条由真正例构成的通路。也就是说，如果不断利用真正例作为额外的查询样本，我们就有很大的可能性，最后查询到假负例，并且将它们的排序提前。据此，我们将初始排序位于顶部的这些图像（准确地说，是图像里的特征）加入检索查询集合。

上述想法很容易用异质图模型来实现。记(6-3)式中排序第一的图像$\mathbf{I}_{0,1}$为$\mathbf{I}_{q,1}$，意味着我们将这一与查询图像最相关的图像用作额外的查询样例。随后，我们更新每一特征的权值为：

$$w(\mathbf{f}_m) = \sum_{r'=0}^{r} \mathbb{I}\left(\mathbf{f}_m \in \mathbf{I}_{q,r'}\right) \tag{6-4}$$

在第1轮扩展中，$r = 1$。随后，(6-2)式就可以用于更新计算图像的分数。

上述迭代过程将进行R轮。在每一轮中，我们在之前没有扩展过的图像候选中，选取当前排序最高的图像进入查询集合，并且相应更新特征的权值和图像的分数。这一算法被称为**增量查询扩展**（Incremental Query Expansion，IQE）。R是算法的**最大扩展轮数**，其作用将在第6.2.3.2节中讨论。

图6.3展示了IQE算法的一个直观例子。查询图像是一幅经过较大修改的*Mona Lisa*图像，其中能够抽取到有效特征的区域很小。显然，利用朴素的检索算法无法获得很好的结果。然而，在经过3轮扩展检索后，我们找到了更多的正例，从而显著地提升了检索结果的召回率。

6.2.2.3　提升准确率：图像特征投票

值得注意的是，IQE算法在提升检索召回率的同时，也可能引入一些假正例，造成检索准确率的下降。图6.3的第3轮扩展查询就是一个例子。为了筛除这些假正例（包括原先就存在的假正例），我们利用图像和特征之间的关系。直接的观察表明，与查询图像相关的图像，往往包含更多相关特征；而相关特征出现的图像，则更有可能是相关的图像。这就构成一个置信度传播模型的基础。我们利用HITS算法[209]设计**图像特征投票**（Image-Feature Voting，IFV）算法来达成这个目的。

在IQE算法结束后，我们已经得到了图像的分数$s(\mathbf{I}_n)$以及特征的权值$w(\mathbf{f}_m)$。IFV算法也分为若干轮，每轮包括四个步骤：(1)图像分数归一化、(2)图像对特征投票、(3)特征对图像投票、以及(4)图像重排序。

图像分数归一化需要对每张图像计算置信函数（belief function）$\Omega(\cdot)$：

$$\Omega(\mathbf{I}_n) = \exp(-\sigma \times \gamma(\mathbf{I}_n)) \tag{6-5}$$

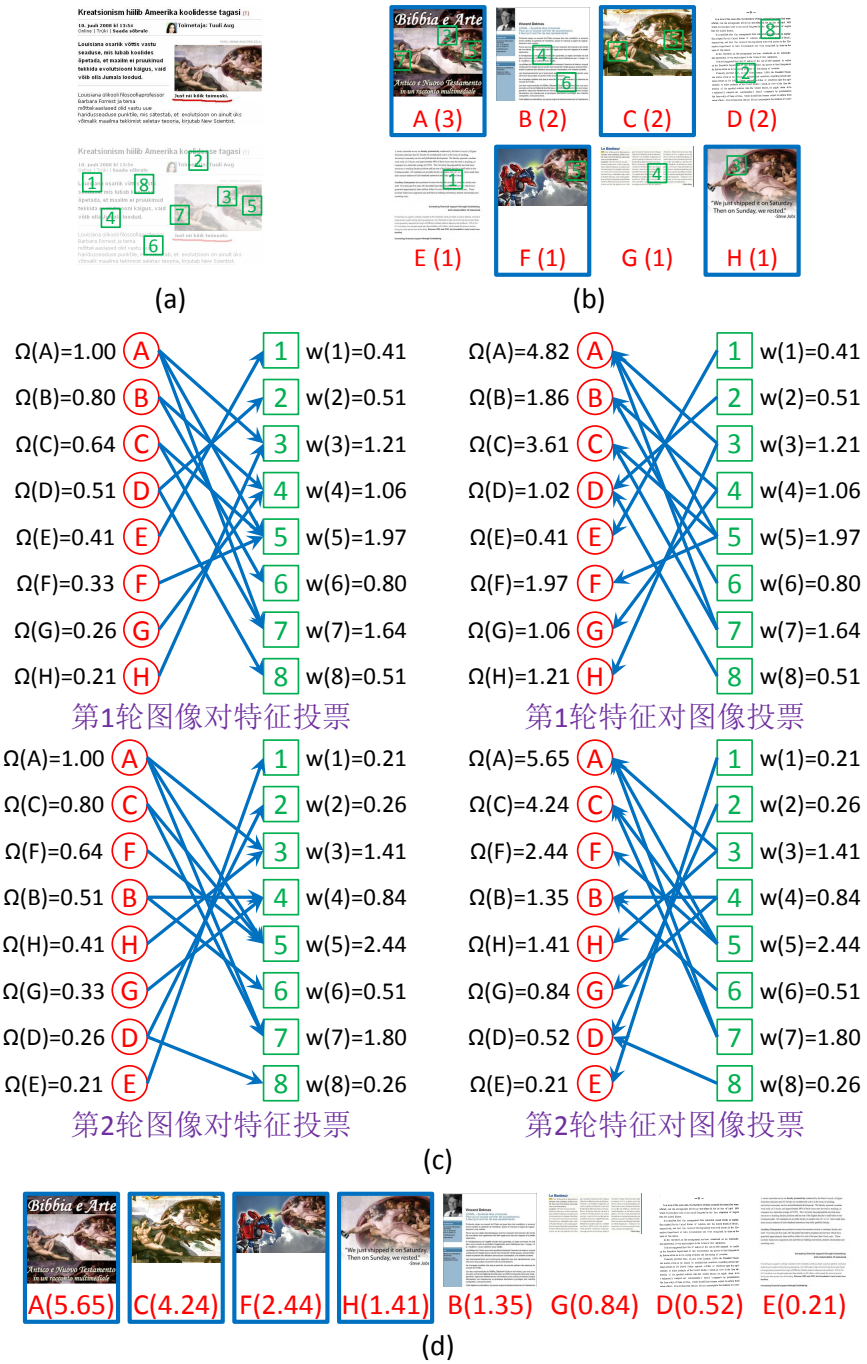


图 6.4　展示图像特征投票（IFV）算法的一个样例。 (a)查询图像包含8个特征，其中3个位于感兴趣区域（油画）上。 (b)初始查询结果，其中蓝框表示真正例。初始结果的mAP值为0.67 $= \frac{1}{4}(1/1 + 2/3 + 3/6 + 4/8)$. (c)两轮图像和特征之间相互投票的过程。一轮过后，检索质量已经明显提升（mAP值为0.95）；两轮过后算法收敛，mAP值为1，表明所有正例都排在负例之前。最后权值最高的3个特征（#5、#7、#3）恰好是位于感兴趣区域的特征。

其中，$\sigma = 0.5$是平滑向量，而$\gamma(\mathbf{I}_n)$图像$\mathbf{I}_n$当前的排名（排在第一名的图像有$\gamma(\mathbf{I}_n) = 1$，而排第二的有$\gamma(\mathbf{I}_n) = 2$，以此类推）。接着，我们用图像对特征进行投票，每个特征的得分就是所有包含它的图像的置信函数之和：

$$w(\mathbf{f}_m) = \sum_{n,\mathbf{I}_n \ni \mathbf{f}_m} \Omega(\mathbf{I}_n) \tag{6-6}$$

反过来，特征对图像也进行投票，每张图像的得分就是所有包含在它之内的特征的得分之和：

$$s(\mathbf{I}_n) = \sum_{m,\mathbf{f}_m \in \mathbf{I}_n} w(\mathbf{f}_m) \tag{6-7}$$

最后，我们根据新一轮的得分对所有图像重新排序。上述过程将持续进行V轮，V是**最大投票轮数**，其作用将在第6.2.3.2节中讨论。

需要强调的是，置信函数$\Phi(\cdot)$对于我们的方法非常重要。它作为归一化函数，不仅可以避免算术溢出，而且还可以使得排名相同的图像始终具有相同的权重，避免了不稳定现象的发生。

图6.4展示了一个例子，包含8张图像和8个量化特征。我们能够看到，IFV算法能够自动地寻找那些有效的特征并且相应地增加它们的权值，从而提高检索的准确率。算法收敛时，权值较高的特征更有可能是重要的特征。

图6.5陈列了IQE算法和IFV算法的详细流程。

6.2.2.4　讨论

我们提出了两个在后处理过程中起到重要作用的算法，即IQE和IFV。直观上说，IQE的目标是寻找查询图像和假负例之间的联系，提升假负例的排名，从而提高检索结果的召回率； IFV则通过在图像和特征之间进行置信度传播，筛除一些假正例，从而提高检索结果的准确率。根据第6.2.3.2节中观察的结果，IQE和IFV都能够显著地提升检索结果的质量。此外根据第6.2.3.3节的结果，IQE和IFV可以很好地配合，因为同时使用两种算法得到的结果比只用其中一种算法得到的结果更好。

由于IQE和IFV算法都包含有量化过程，而且量化方式是人工定义的（如置信函数(6-5)），对计算过程进行定量分析（如收敛性、收敛速度等）就变得非常困难。然而根据HITS算法[209]的定性分析结果，只要排名靠前的图像或者特征在量化过程中被赋予更高的权值，那么收敛性在大多数情况下都能够得到保证。其他基于随机游走的算法[210] [24]也提供了类似情况下收敛性的数学证据。在实验中（见第6.2.3.2节），每一个查询样例都以收敛结束，也说明了算法的可靠性。

算法：异质图传播（HGP）

1. **输入：**
   图像集合$\mathcal{I} = \{\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_N\}$，特征集合$\mathcal{F} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_M\}$，查询图像$\mathbf{I}_{q,0}$，超参数$R$、$V$以及$\sigma$。

2. **增量查询扩展（IQE）：**
   - 利用(6-1)式，初始化特征$\mathbf{f}_m$的权值$w(\mathbf{f}_m)$；利用(6-2)式，初始化图像$\mathbf{I}_n$的分数$s(\mathbf{I}_n)$。
   - 对于$r = 1, 2, \ldots, R$，找到相应的扩展图像$\mathbf{I}_{q,r}$，分别利用(6-4)式和(6-2)式迭代更新特征权值$w(\mathbf{f}_m)$和图像分数$s(\mathbf{I}_n)$。
   - 重新排序，得到查询扩展后的检索结果。共进行R轮扩展。

3. **图像特征投票（IFV）：**
   - 将IQE产生的结果作为初始排序。
   - 利用(6-5)式计算图像的置信函数，并且分别利用(6-6)式和(6-7)式更新特征权值和图像分数。
   - 在每一轮结束后，根据分数将图像重新排序。共进行V轮投票。

4. **输出：**
   重新排序后的检索结果，即IFV最后一轮之后的图像排序。

图 6.5　IQE和IFV算法的流程。

6.2.2.5　算法的加速

现在，我们分析算法的复杂度。算法的计算开销由三个部分组成：初始检索过程、IQE和IFV；其中，初始检索过程的复杂度取决于基准检索系统。

对于IQE算法，我们需要进行R轮额外检索（使用R张扩展的图像），这就需要花费相当于基准算法R倍的时间。使用恰当的技巧，牺牲一定的检索精度，就可以显著地降低额外检索所花费的时间。比如对于向量量化（Vector Quantization）算法如[6][102]，我们可以只利用额外查询图像中20%最经常出现的特征（停用词除外）；对于标量量化算法[83]，将其中的码本扩展阈值（codeword expansion threshold）从$d = 2$降低为$d = 1$，就可以将每次扩展查询的时间开销降低至大约20%。通常IQE算法需要进行大约10轮（见第6.2.3.2节）。利用上述手段，我们

额外花费的时间复杂度就是初始检索过程的大约2倍，这是可以接受的。

IFV算法的每一轮需要将置信值在图像和特征之间来回传播。假设数据集共有N张图像，而特征集合中共有M个特征，那么一个V轮的投票算法就需要$2VNM$次代数运算。在大规模网络图像搜索任务（见第6.2.3.1节）中，典型设定为$N = 10^6$、$M = 500$、$V = 5$（见第6.2.3.2节）。这样，总共的浮点计算次数大约为5×10^9，在单个3.0GHz的CPU上耗时约8秒。为了加速，我们可以相应减少进入重排序过程的图像数量：只抽取N张图像中的前U张进入重排序。这里U被称为**最大投票候选数**。当$U \ll N$时，总共的计算量就大幅降低为$2VUM$次浮点运算。根据第6.2.3.2节的实验结果，在一个包含1百万张图像的数据集里，只选取$U = 1000$张排序靠前的图像进行重排序，就可以大幅地改进检索结果，同时计算时间也显著地减少为大约100毫秒。

6.2.2.6　与之前方法的对比

这里，有必要将我们的算法与此前的后处理方法进行对比。我们分别将IQE算法与查询扩展、IFV算法与基于扩散的重排序算法对比。

IQE算法（增量式的查询扩展）与之前的方法[23][104][211]有显著的不同。此前的方法往往一次性地选取扩展出的特征，而IQE允许我们在每次迭代中选择一个最有可能的正例用于扩展。这样做的好处是增加了扩展图像是正例的概率。回到图6.3所示的实例，如果我们简单地使用初始的前10张图像用于查询扩展，其中只有3张图像是正例；而IQE算法执行10轮后，扩展的图像中有7张是正例。我们还在整个数据集上统计了这一数据，发现IQE算法让前10次扩展的平均正例图像比例从37%增加到了75%。这表明我们找到了更有效的扩展特征，提升了结果的召回率，同时也避免了引入过多的噪声，影响结果的准确率。

另一方面，IFV算法在图像和特征之间进行投票，也是一个显著的创新点。与之前的方法[108][24][121]对比， IFV算法具有简单（可直接在任何查询结果上开始重排序）、稳定性（收敛性由随机游走理论保证[209]）以及可扩展性（可以很容易地将这种易并行的算法推广到实际大小的数据集中）的优势。然而，由于IFV的前提条件之一是我们必须拥有足够数量的正例（见第6.2.3.6节），它并不适用于诸如**UK-Bench**[74]（其中每张查询图像只有4个正例）的数据集。幸运的是，IQE和IFV都能够在大规模网络图像搜索的假设下很好地工作。

另一个IQE和IFV算法与传统方法[106][118]的区别，在于我们的算法无需使用特征的几何信息，如位置、尺度、朝向，等等。正如第6.2.3.5节所述，这使得算法所需的时间和空间开销都有效地降低了。虽然几何信息的丢失使得我们无法进行空间验证，但实验结果表明IFV算法也能够有效地筛除空间不吻合的负例。对

比于以往的无需几何信息的算法[116][211]，我们报告的检索质量更高一些（参见第6.2.3.3节）。相比于和我们的算法非常类似的方法[211]，我们使用了不同的扩展和重排序方式，这使得我们的检索结果更好，算法的运行速度也更快。此外，我们还提出使用异质的图结构，这与[212]中提出的同质图结构不同。虽然异质图结构也曾经被用于类似的任务[213]，但是我们的方法创新性地提出将图像和特征都作为显式节点：这样，在获得图像检索结果的同时，也能挖掘出更重要的特征信息。

最后，我们的算法的形式是非常一般化的。这使得我们的算法很容易实现，应用范围很广（几乎能够与所有基于局部特征的检索算法配合），甚至可以用在其他的后处理方法之后，进一步地提升图像检索质量。

6.2.3　实验部分

6.2.3.1　数据集和实现细节

我们在四个常用图像检索数据集上进行两类实验。

第一部分是大规模近似近邻网络图像搜索。我们使用两个数据集：**DupImage**数据集[107]包含33组近似近邻概念，共计1104张图像；**CarLogo-51**数据集[110]包含51种知名汽车的标志，共计11903张图像。我们还从网络上随机抓取1百万张图像作为无关样本，以测试算法的可扩展性。为了测试算法表现与无关样本数量之间的关系，我们还在其中随机选取了3个子集，分别包含10万、20万和50万张无关图像。所有图像首先被重置大小，在保证其长宽比不变的情况下，将其长边重置为300个像素。我们在DoG检测子[4]发现的兴趣区域上，抽取RootSIFT特征[118]，并且使用标量量化（SQ）算法[83]将描述子编码为256位的0/1向量。256位中的前32位用作倒排表的索引地址，而后$224 = 256 - 32$位则与图像ID一起存储在倒排表中。出现在超过1%图像中的视觉单词被视为停用词。在线检索时，我们使用码本扩展阈值（codeword expansion threshold）$d = 2$（在IQE过程中$d = 1$，见第6.2.2.5节）和海明阈值（Hamming threshold）$\kappa = 24$。

第二部分是（略小规模）近似重复物体检索。我们使用两个数据集：**Oxford buildings**（**Oxford5K**）数据集[6]，包含5063张图像和55个查询；以及**Paris buildings**（**Paris6K**）数据集[8]，包含6391张图像和55个查询。两个数据集都与Flickr上随机抓取的10万张无关图像混合，形成**Oxford105K**和**Paris106K**数据集。所有图像不被重置大小，在Hessian Affine检测子[61]发现的兴趣区域上，抽取RootSIFT特征[118]，并且利用近似K-Means算法[6]训练大小为1百万的码本。在特征进行硬量化后，我们使用传统倒排表建立索引，并且使用ℓ_p范数IDF[102]计算特征的权值。

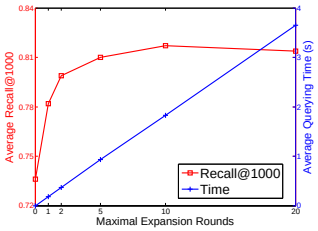


图 6.6　平均的1000位召回率和平均查询时间（s）与IQE算法参数R的关系。此处并未涉及IFV算法。

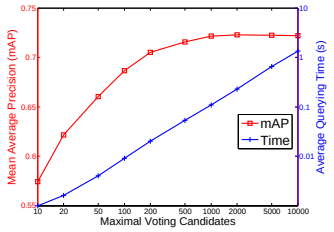


图 6.7　平均的mAP值和平均查询时间（s）与IFV算法参数U的关系。此处，固定$R = 10$以及$V = 5$。

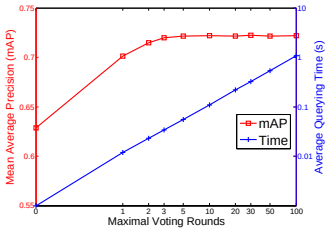


图 6.8　平均的mAP值和平均查询时间（s）与IFV算法参数V的关系。此处，固定$R = 10$以及$U = 1000$。

出现在超过1%图像中的视觉单词被视为停用词。

注意到，在两部分实验中，基准算法[83][102]是不同的。这可以说明我们的后处理模块能够与不同的算法配合，达到更好的检索结果。

6.2.3.2　参数的影响

我们研究算法中参数的影响，包括最大扩展轮数R、最大投票候选数U和最大投票轮数V。对于R，我们测试1000位召回率（由IQE算法产生）；而对于U和V，我们测试mAP值（由两个算法共同产生）。所有的结果都在**DupImage**数据集与1百万无关样本混合后进行测试。图6.6、 6.7和6.8分别展示了三个参数的作用。我们可以发现，IQE和IFV都显著地提升了图像检索的精度。

我们还对不同的参数，记录了每张查询图像的平均检索时间，并且将它们画在同一张图像中。可以看到，图像检索的时间开销几乎随着R、U和V的增加线性增长；而更大的参数，例如$R = 20$和$V = 10$，对于检索精度的提升已经非常有限。类似的现象也能够在其他几个数据集的实验中观察到。因此，我们在精度和复杂度之间选取平衡点，选取$R = 10$、$U = 1000$和$V = 5$，用于后面的实验。当然，上述参数也可以通过自动选择方法（如检验算法的收敛性）进行选择，不过我们强调，这些人工挑选的参数在所有数据集上都取得了良好的效果。对于每个数据集，参数组$(R = 20, V = 10)$与$(R = 50, V = 50)$之间的mAP差距小于0.01。

6.2.3.3　算法的表现

HGP算法和之前的一些方法在大规模近似重复网络图像搜索任务上的检索结果陈列在表6.1和表6.2中。在**DupImage**和**CarLogo-51**数据集与1百万无关样本混合后，我们的方法分别取得了0.72和0.30的mAP，与基准算法SQ[83]相比，产生的相对提升分别为33.2%和42.9%。

数据集大小	100K	200K	500K	1M
Nister[74]	0.5093	0.4727	0.4233	0.3776
Jegou[75]	0.5526	0.5125	0.4646	0.4287
Philbin[8]	0.6401	0.5963	0.5317	0.4804
Zhou[107]	0.6071	0.5868	0.5570	0.5294
SQ[83]	0.6289	0.6067	0.5736	0.5417
SQ + IQE	0.7643	0.7472	0.6971	0.6659
SQ + IFV	0.6873	0.6698	0.6469	0.6202
SQ + IQE + IFV	**0.8038**	**0.7822**	**0.7504**	**0.7216**

表 6.1　**DupImage**与不同数量无关样本混合后，各方法取得的mAP值。

数据集大小	100K	200K	500K	1M
Nister[74]	0.1860	0.1755	0.1610	0.1500
Jegou[75]	0.2165	0.2036	0.1865	0.1733
Philbin[8]	0.2448	0.2326	0.2149	0.2004
SQ[83]	0.2449	0.2350	0.2238	0.2109
SQ + IQE	0.3187	0.3074	0.2954	0.2812
SQ + IFV	0.2819	0.2693	0.2578	0.2446
SQ + IQE + IFV	**0.3431**	**0.3314**	**0.3187**	**0.3014**

表 6.2　**CarLogo-51**与不同数量无关样本混合后，各方法取得的mAP值。

我们同时报告HGP算法与此前的一些算法在近似重复物体检索问题上的结果。结果陈列于表6.3中。能够看到，基准算法[102]产生的检索质量在后处理过程中被显著提高。而且，我们的算法比同样不使用几何特征的算法[116]产生的结果更好，与[211]的结果可比。尽管比最好的精度[106]与[118]略低，但是我们的算法更快，且不使用几何特征，从而能够有效地降低时间和空间开销（见第6.2.3.5节）。同时我们强调，HGP算法也能够应用于其他方法之后，进一步提升后处理的效果。

基于上述两个实验，我们能够得出结论：HGP算法是一个有效、可扩展的图像检索后处理算法。这种算法能够应用于不同的任务上，也能够与不同的基准算法相配合。

6.2.3.4　样例搜索结果

图6.9展示了在**DupImage**数据集与1百万无关样本混合后的一个典型查询样例。如图所示，由于查询图像被噪声干扰，许多正例与查询图像之间都无法找

数据集	Oxford5K	–105K	Paris6K	–106K
Philbin[6]	0.647	0.541	–	–
Philbin[8]	0.825	0.719	0.718	0.605
Chum[104]	0.827	0.767	0.805	0.710
Perdoch[106]	0.916	–	0.885	–
Mikulik[214]	0.849	0.795	0.824	0.773
Qin[116]	0.814	0.803	0.767	–
Arandjelovic[118]	**0.929**	**0.891**	**0.910**	–
Toilas[211]	0.838	0.804	0.828	–
Toilas[211]+SP	0.880	0.840	0.828	–
ℓ_p-IDF[102]	0.746	0.704	0.622	0.594
ℓ_p-IDF + IQE	0.825	0.806	0.789	0.740
ℓ_p-IDF + IFV	0.793	0.776	0.702	0.685
ℓ_p-IDF + IQE + IFV	**0.877**	**0.858**	**0.845**	**0.812**

表 6.3　在**Oxford5K**和**Paris6K**数据集（可能混合**Flickr100K**数据）上，各方法取得的mAP值。

数据集大小	100K	200K	500K	1M
基准算法[83]	77	132	276	477
IQE（我们的工作）	150	255	532	932
IFV（我们的工作）	18	31	59	110
HGP（基准+IQE+IFV）	245	418	867	1519

表 6.4　我们的算法在**DupImage**和**CarLogo-51**数据集（混合不同数量的无关样本）上的总时间（毫秒）开销。 SQ算法[83]被用于基准算法（和查询扩展算法）。计算中，我们使用单个3.0GHz的CPU核。

到3个以上的特征匹配。我们使用的两个基准算法[74][83]对于这些样本都无能为力。然而通过IQE和IFV，我们成功地找回了许多困难的正例，并且筛除了大量的负例。在该查询样例上，使用HGP算法得到的mAP值是0.952，显著超过了两个基准算法：[74]的0.337和[83]的0.521。

6.2.3.5　时间和空间开销

关于时间开销的理论分析见第6.2.2.5节。本节的实验都是在一个8核、主频为3.0GHz的CPU上进行的。

大规模近似近邻图像搜索以及物体检索的时间开销分别列在表6.4和表6.5中。所有的例子都使用了第6.2.2.5节中讨论的加速方法。可以看出，我们的算法是高

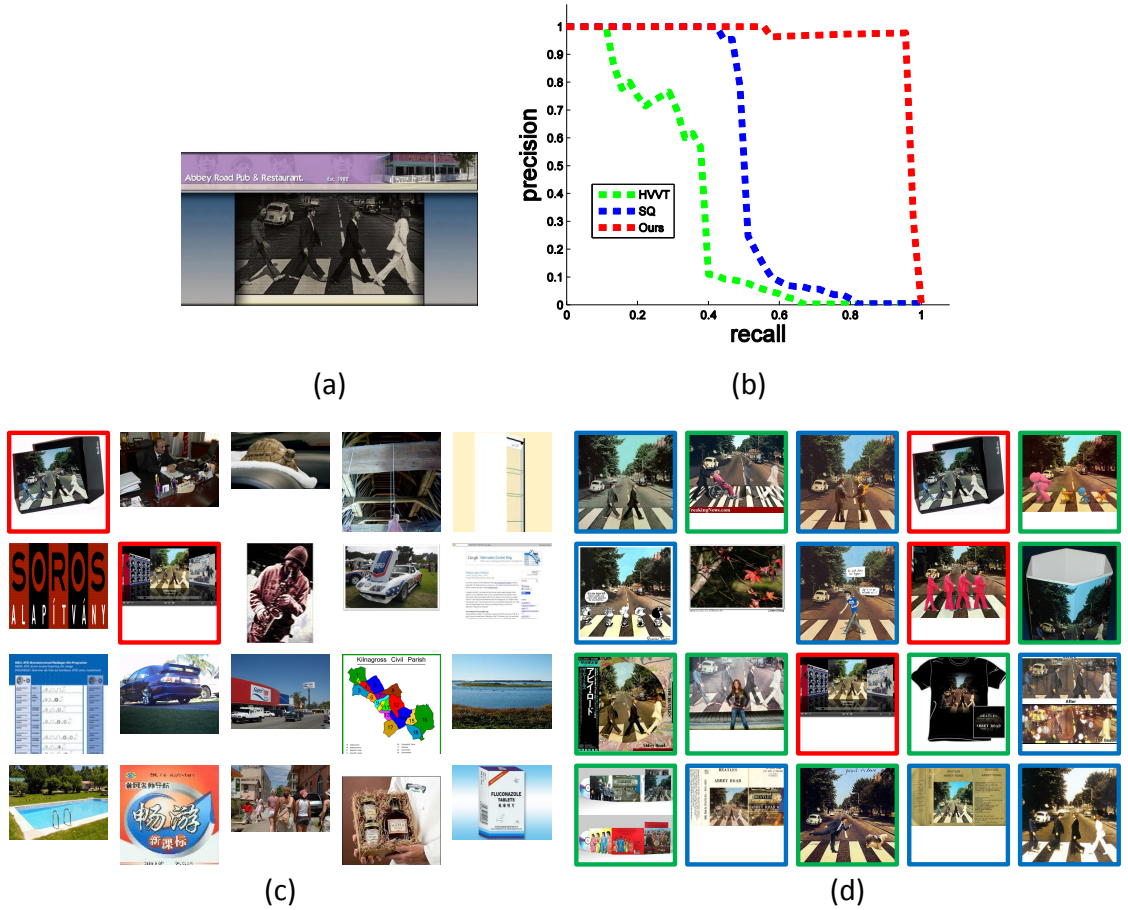


图 6.9　在**DupImage**数据集与1百万无关样本混合后，一个典型的查询结果。 (a)一个困难的例子，其中感兴趣区域被随机噪声所影响。 (b)HVVT算法[74]、SQ算法[83]以及我们的方法产生的准确率-召回率曲线。我们在mAP值（即准确率-召回率曲线下方的面积）上得到了显著的提升。 (c)(d)SQ算法与我们的方法查询结果的第21到第40名图像（由于两种方法的前20名图像都是真正例，我们就不再展示这些图像）。其中，真正例用带有颜色的框表示：红框表示查询图像与候选图像有3个或者更多的匹配特征；蓝框表示有1个或者2个匹配特征；而绿框则表示没有匹配特征。显然，蓝框和绿框标定的图像很难在基准检索过程中被找到。我们的方法利用一系列的真正例进行扩展查询，最终找回了这些困难的样本。

度可扩展的，因为它的时间开销与数据集的大小呈次线性（sub-linear）关系。对于混合了1百万图像的**DupImage**和**CarLogo-51**数据集，图6.10展示了单一查询的时间开销。对于**Oxford105K**和**Paris106K**数据集，我们的算法大约需要450毫秒处理一个查询任务，明显快于一些复杂的后处理算法[106][118]（通常需要超过1秒）。对比于类似的几何无关的后处理方法[211]（需要955毫秒处理一张查询图像），我们的方法得到的结果更好一些，而且只需要不到一半的时间开销。

VQ算法配合典型的倒排表结构，需要在每个特征花费4字节（用以存储图

数据集大小	5K	105K	6K	106K
基准算法[102]	23	124	34	139
IQE（我们的工作）	48	256	71	270
IFV（我们的工作）	7	30	10	36
HGP（基准+IQE+IFV）	78	410	115	445

表 6.5　我们的算法在**Oxford5K**和**Paris6K**数据集（可能与**Flickr100K**混合）上的总时间（毫秒）开销。ℓ_p范数IDF[102]被用于基准算法（和查询扩展算法）。计算中，我们使用单个3.0GHz的CPU核。

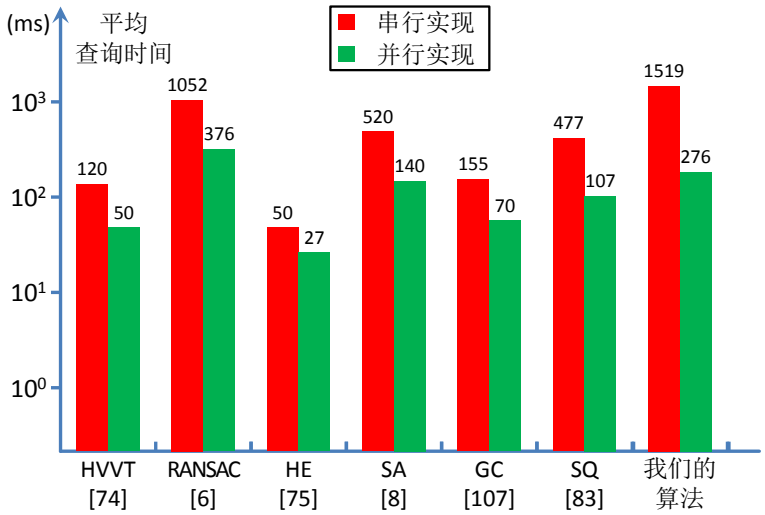


图 6.10　在**DupImage**数据集混合1百万无关样本上，平均的查询时间开销（毫秒）。这里的时间开销并不包括特征的抽取、量化和离线索引过程。我们使用8个核进行并行计算。

像ID）[6]，而SQ算法配合典型的倒排表结构，需要在每个特征花费32字节（用以存储图像ID以及剩余的0/1编码）[83]。如果加入几何信息，即使是最简单的特征坐标信息，每个特征也至少需要额外花费4字节（x和y坐标分别需要2字节）。这样，我们的几何无关的算法相比于几何相关的算法，就在VQ和SQ的基础上分别节省了50%和11%的存储空间。

总之，我们的方法得到的检索精度显著超过了两个基准算法，而且只花费了大约3倍的时间开销。与复杂的后处理算法[106][118][211]对比， HGP算法的效率更高，而且产生的检索结果也是可比的。因此，我们的算法可以被很方便地移植到大规模网络图像检索任务中。

6.2.3.6　提升来源分析

最后，我们展示一个有趣的分段对比实验。我们记录在**DupImage**数据集

SQ所得mAP	占百分比	HGP算法的平均mAP收益
0.0 – 0.2	23.4%	−0.0572
0.2 – 0.5	21.5%	+0.4327
0.5 – 0.8	23.4%	+0.3242
0.8 – 1.0	31.7%	+0.0769
总计	100.0%	+0.1799

表 6.6　我们的算法在SQ[83]基础上，对mAP提升的分段统计。这些结果是在**DupImage**数据集与1百万无关样本混合时得到的。

与1百万无关样本混合后，SQ算法[83]产生的初步检索结果。我们将查询图像按照基准mAP值分为四类，并且计算这四类查询在使用了HGP算法后，所获得的mAP提升。从表6.6展示的结果中可以看出，HGP算法对总体mAP值的提升，主要来自于中等难度的查询图像（初始mAP值在0.2到0.8之间）。对于非常困难的查询（mAP值小于0.2）我们的算法反而稍微降低了检索精度，因为我们的假设此时已经不再满足：当排名靠前的样例中包含更多负例而不是正例时，不论是IQE还是IFV，都无法继续提升检索质量。因此，HGP算法也就不适用于**UKBench**数据集[74]（每个查询只有4个正例，却有大量负例）。但在大规模网络图像检索任务中，我们的算法通常能够表现得很好，因为网络上通常存在大量相似的图像。

6.2.4　结论

本节的主要创新点集中在图像检索的后处理模块。我们使用一个基于图结构的视角观察后处理，并且提出了一个异质图结构，显式地对图像和特征进行建模。两个后处理模块，增量查询扩展（IQE）和图像特征投票（IFV），分别提升了检索结果的召回率和准确率。由于我们的算法不需要使用几何信息，时间和空间开销都得到了显著的降低。实验结果表明，HGP算法能够在两类稍有不同的检索任务上工作，并且与不同的基准算法（VQ和SQ）很好地配合。相比于那些复杂的后处理方法[106][118]，HGP算法更简单高效，而且更加通用：它几乎可以与任何基于局部特征的检索模型配合，提升初始检索精度。

6.3　图像网络算法

6.3.1　问题综述

尽管基于视觉词袋模型的图像检索流程简单高效且可扩展，但是它们产生的结果通常不能达到很高的准确率（precision）或召回率（recall）。其中，主要的原

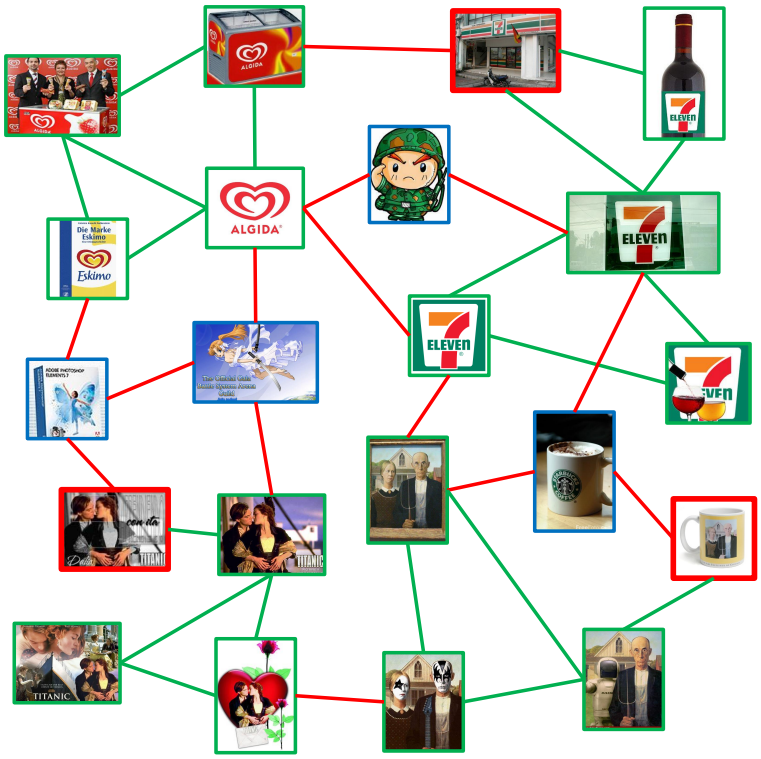


图 6.11　一个简单的例子，展示了图像检索结果中可能出现的负例（用蓝框标记）。拥有不少于10个公共特征的图像对之间连有边，其中绿边和红边分别表示正确的和错误的图像关系。在这样一个充满噪声干扰的图结构中，简单的视觉词袋模型很难获得满意的检索结果，尤其对于那些困难的查询图像（用红框标记）。

因包括局部特征（如SIFT）的描述力不足，以及在量化过程中产生的信息丢失。事实上，局部特征之间的匹配往往对于图像变换（如人工编辑、几何拉伸和变形等）比较敏感，从而导致许多情况下，相关的特征无法被量化到同一个单词，而不相关的特征却被错误地匹配在一起。上述现象导致的直接结果，就是在查询过程中，相关的图像被排序在不相关的图像之后，导致检索质量的显著下降。为了解决上述问题，后处理（post-processing）模块被广泛地应用于图像检索算法中，利用图像的额外信息（如特征的几何信息[75]、初始排序靠前的图像中提取的额外特征[23] 以及置信传播[24]），对检索结果进行重排序，从而达到提升检索质量的目的。尽管所有这些方法都能有效地提升检索结果的准确率和召回率，然而手工设计的后处理方法依然限制了它们的可扩展性。

本节同样以图结构为线索来考虑图像检索的后处理问题。在上一节介绍的HGP算法的基础上，我们考虑一种更加自然而通用的图像重排序算法，其中不包含任何人工设计的技巧。首先，我们对图6.11所示的一个由图像组成的小型图结构进行观察。在这个图结构中，每个节点都是一张图像，两个节点之间有边相

连，当且仅当对应图像之间有不少于10个的公共特征。可以看到，由于局部特征的不稳定性，图中存在一些没有连边的相关图像，也存在一些连边的不相关图像。直觉判断，前者会造成检索的召回率下降，而后者则会影响检索的准确率。为了克服这样的缺陷，我们提出：图像之间的匹配远比特征之间的匹配更加稳定。同时，下面的这些观察结果将是提升检索质量的关键：

- 对于大部分无法直接找到的真正例，我们可以利用图结构中一系列的连边（由真正例组成的路径）找到它们。
- 对于大部分的查询，结果中真正例的数量都多于假正例的数量。因此，采用类似于多数投票（majority voting）的方式就能够有效地将假正例筛除。

可以看出，这些观察与上一节的HGP算法非常类似。基于这些观察，我们提出**图像网络（ImageWeb）**，一种新的数据结构，用于改进图像级别的检索精确度。从本质上说，ImageWeb是一个稀疏的图结构，其中每个节点表示一张图像。从图像$\mathbf{I}_a$到图像$\mathbf{I}_b$存在有向边连接，当且仅当图像$\mathbf{I}_b$出现在图像$\mathbf{I}_a$查询结果的前列。这样，图结构中的连接就表示了“$\mathbf{I}_a$认为$\mathbf{I}_b$相关”这一信息，因此我们可以利用基于随机游走（random walk）的算法，如HITS[209]，进行置信传播，从而对检索结果进行重排序。实验结果表明，通过这样的迭代计算，可以非常有效地提升检索结果的精度。

本节的主要贡献总结可以总结为以下两点。第一，我们提出了一种有效的数据结构，即图像网络（ImageWeb），用以建模图像级别的关系；同时我们设计了有效的创建、插入和删除算法，使得ImageWeb结构能够有效地应用于实际问题中。第二，我们提供了一种基于精度和速度平衡的准则，以指导参数选择过程。最终得到的系统，能够在仅仅花费基准算法20%查询时间的情况下，取得比基准算法高得多的检索精度。

6.3.2　ImageWeb数据结构

本节阐述ImageWeb数据结构，用以进行图像重排序，提升初始检索的质量。我们从文本检索的一些常见概念出发，构建有效的数据结构，并完善相应的算法。随后，我们将分析ImageWeb的时间和空间开销。

6.3.2.1　文本检索和链接分析

文本搜索引擎在互联网中扮演着重要的角色。一个典型的文本搜索引擎通常包含三个部分。首先，一个网络爬虫（Web crawler）将沿着网页上的链接寻找尽可能多的页面，并且将它们存储在本地索引（index）中。当用户输入查询词时，

搜索引擎查看本地索引，并且将那些包含查询词的网页返回给用户。虽然包含给定单词的网页可能有成千上万，但其中的一些往往比另一些更加相关（或者更符合用户需求）。于是此时，搜索引擎使用链接分析（link analysis）技术，对网页进行重排序，以保证靠前的网页尽可能更加贴合用户的需求。

链接分析是一种常用的技术，它将互联网看成一个由网页节点构成的图结构，并且利用随机游走（random walk）模型模拟用户行为。链接分析的目标是计算每个网页的置信度（affinity value）[215]。它基于的思想很简单：高质量的网页更有可能包含指向高质量网页的链接，因此任一网页的质量都可以通过指向它的网页的质量近似估计。 PageRank[210]和HITS[209]算法都是基于这一思想设计的。它们既可以通过迭代算法不停地计算，也可以通过计算图结构的本征值（eigenvalues）来进行分析。

PageRank为每个网页赋予一个权值$w \in (0,1)$，代表它在浏览过程中被用户访问到的概率。在迭代过程中，每个网页的权值会被散发到它指向的网页中去，而它的新权值则通过收集指向它的网页散发的权值而获得。我们可以简单地设置所有网页具有同样的初始权值。 PageRank的最终结果可以反映网络的本质结构，因此它是查询独立（query independent）的排序算法。 HITS算法（又称为枢纽值和权威值（Hubs and Authorities）算法）同样为每个网页节点分配数值。不同的是，每个节点被分配两个值$h, a \in (0,1)$，即枢纽值和权威值，分别表示该网页能够被访问到的概率和该网页链接的可信程度。在每次迭代过程中，我们首先通过散发枢纽值得到权威值，再通过散发权威值得到枢纽值。与PageRank算法不同的是，HITS算法是查询相关（query dependent）的。它从某个查询开始计算，被查询网页的枢纽值和权威值被初始设定为1，而其他网页设定为0。迭代完成后，HITS算法能够反映从给定网页出发，到达其他网页的概率。

6.3.2.2　ImageWeb：定义和构建

文本检索的链接分析算法基于这样一个假设：网页$\mathbf{P}_a$有链接指向网页$\mathbf{P}_b$，说明$\mathbf{P}_a$认为$\mathbf{P}_b$质量高。为了将链接分析应用于图像检索中，我们也将图像组织为一个图结构，其中每个节点代表一张图像。从图像$\mathbf{I}_a$对应的节点到图像$\mathbf{I}_b$对应的节点存在有向边，当且仅当$\mathbf{I}_b$位于$\mathbf{I}_a$检索结果的前若干位。这样，同样的假设就可以推广到图像中：$\mathbf{I}_a$“链接”到$\mathbf{I}_b$，说明$\mathbf{I}_b$更有可能与$\mathbf{I}_a$相关。

将上述想法表达出来，就形成了ImageWeb的雏形。 ImageWeb是用于描述图像层次关系的数据结构。假设我们有一个数据集$\mathcal{I} = \{\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_N\}$，其中$N$是图像数目（在实际应用中可达1百万或者更大）。 **图像网络**（**ImageWeb**）就定义为一个有向图$\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$。其中，$\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$是节点集合，$v_n$是对应于图像$\mathbf{I}_n$的

算法1：创建ImageWeb

1. **输入：**
   一个图像数据集：$\mathcal{I} = \{\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_N\}$，ImageWeb宽度$K$。

2. **索引和初始化：**
   将$\mathcal{I}$中包含的所有特征放入一个倒排表。初始化节点：$\mathcal{V} = (v_1, v_2, \ldots, v_N)$，其中$v_n$是对应于图像$\mathbf{I}_n$的节点，$n = 1, 2, \ldots, N$。

3. **链接构建：**
   对于每个v_n，将$\mathbf{I}_n$作为查询，在倒排表中进行检索。在v_n和查询结果前K名图像对应的节点之间建立有向边，边权为两者共有（匹配）的特征个数。

4. **归一化：**
   每个节点的出边权值被归一化，使得它们的和为1。

5. **输出：**
   一个ImageWeb，表示为：$\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$。

图 6.12　ImageWeb：创建网络的一般步骤。

节点，$n = 1, 2, \ldots, N$；而$\mathcal{E}$是节点间的边的集合，从v_{n_1}到v_{n_2}有一条有向边，当且仅当$\mathbf{I}_{n_2}$出现在$\mathbf{I}_{n_1}$初始查询结果的前K位中。这里超参数K是一个远小于N的整数，也被称为ImageWeb结构的**宽度**。 ImageWeb的每一条边都有一个权值，表示它的重要程度：权值越大，表示连边的两张图像越有可能相关。任何时刻，任一节点出边的权值都经过归一化，即它们的权值和为1。图6.12展示了构建ImageWeb的算法。

在实际应用中，图像数据集可能会随着时间发生变化。新索引到的图像可能会加入数据集，而某些过时或者非法的图像也会被从数据集中移除。因此，有必要设计算法对ImageWeb结构进行插入和删除操作。插入和删除操作的主要困难来源于维护那些原本存在（仍需保留）于数据集中的图像，使得它们的出边能够适应新的数据集。显然，如果每次插入和删除操作后都重新计算整个数据集图像的出边，那么计算开销可能会非常大。我们采用一种近似的方法：只检查那些与插入或删除图像相关的图像的出边，并且进行必要的修改。在运行一段相对长的时间后，我们可以一次性地重建整个ImageWeb，以降低均摊（amortized）的计算复杂度。图6.13和6.14分别描述了ImageWeb的插入和删除过程。

算法2：向ImageWeb中插入图像

1. **输入：**
   原始ImageWeb结构$\mathcal{G}$，一个新的图像数据集：$\mathcal{I}^{\mathrm{i}} = \left\{\mathbf{I}_1^{\mathrm{i}}, \mathbf{I}_2^{\mathrm{i}}, \ldots, \mathbf{I}_{N_{\mathrm{i}}}^{\mathrm{i}}\right\}$。

2. **索引和初始化：**
   将$\mathcal{I}^{\mathrm{i}}$中包含的所有特征加入原先的倒排表。初始化节点：$\mathcal{V}^{\mathrm{i}} = \left(v_1^{\mathrm{i}}, v_2^{\mathrm{i}}, \ldots, v_{N_{\mathrm{i}}}^{\mathrm{i}}\right)$，其中$v_n^{\mathrm{i}}$是对应于图像$\mathbf{I}_n^{\mathrm{i}}$的节点，$n = 1, 2, \ldots, N_{\mathrm{i}}$。

3. **$\mathcal{I}^{\mathrm{i}}$中的链接：**
   对于每个v_n^{i}，将$\mathbf{I}_n^{\mathrm{i}}$作为查询，在倒排表中进行检索。在$v_n^{\mathrm{i}}$和查询结果前$K$名图像对应的节点之间建立有向边，边权为两者共有（匹配）的特征个数。

4. **更新$\mathcal{V}$的出边：**
   对于每个v_n^{i}，检查它的前K名查询结果：如果存在某个v_k'，使得v_k'和v_n^{i}之间的匹配权值大于v_k'的第K个出边权值，那么将v_n^{i}插入v_k'的出边的合适位置，并且将v_k'的最后一个出边删除。

5. **归一化：**
   每个节点的出边权值被归一化，使得它们的和为1。

6. **输出：**
   更新后的ImageWeb：$\mathcal{G}'$。

图 6.13　ImageWeb：插入图像的一般步骤。

6.3.2.3　ImageWeb：链接分析

图6.15展示了一个小型ImageWeb（$N = 1104$）内的两个节点（$K = 10$）。我们分别展示了一个较简单和一个较困难的查询样例。值得注意的是，相比于图像数据集的大小来说，K通常是非常小的一个数，因此ImageWeb通常无法存储一个图像的所有正例。 ImageWeb所包含的信息是一些经过预先计算的检索结果，它们可以用于后处理环节，提升图像检索质量。

ImageWeb上的链接分析算法有两个目的：寻找那些没有直接链接的正例、筛除那些位于初始查询结果前列的负例。我们充分利用一个事实：所有正例图像都包含相同的近似重复概念，从而很容易在它们之间建立起一个网络链接。这样，就可以通过扩散置信值来寻找更多的正例，同时也能通过多数投票的机制来筛除负例。以图6.15中较为困难的例子说明，虽然初始检索结果的前10名中有6个负

算法3：从ImageWeb中删除图像

1. **输入：**
   原始ImageWeb结构$\mathcal{G}$，删除图像集合$\mathcal{I}^{\mathrm{d}} = \left\{\mathbf{I}_1^{\mathrm{d}}, \mathbf{I}_2^{\mathrm{d}}, \ldots, \mathbf{I}_{N_{\mathrm{d}}}^{\mathrm{d}}\right\}$。

2. **索引和初始化：**
   将$\mathcal{I}^{\mathrm{d}}$中包含的所有特征从原先的倒排表中删除。删除这些节点：$\mathcal{V}^{\mathrm{d}} = \left(v_1^{\mathrm{d}}, v_2^{\mathrm{d}}, \ldots, v_{N_{\mathrm{d}}}^{\mathrm{d}}\right)$，其中$v_n^{\mathrm{d}}$是对应于图像$\mathbf{I}_n^{\mathrm{d}}$的节点，$n = 1, 2, \ldots, N_{\mathrm{d}}$。

3. **更新$\mathcal{V}$的出边：**
   对于每个v_n^{d}，检查所有的节点：如果存在某个v_k'，使得v_n^{d}出现在v_k'的前K个查询结果中，那么将v_n^{d}从v_k'的出边中删除。

4. **检查：**
   对于所有保留下来的图像，检查它们的前K个查询结果剩余的个数。如果数量低于某个阈值，如$0.8K$，那么为这张图像重新检索，建立新的出边。

5. **归一化：**
   每个节点的出边权值被归一化，使得它们的和为1。

6. **输出：**
   更新后的ImageWeb：$\mathcal{G}'$。

图 6.14　ImageWeb：删除图像的一般步骤。

例，但是我们注意到这6个负例事实上来自于6个不同的近似重复组，它们之间并没有直接关系，因此很容易被后处理算法筛除。

上述观察引导我们应用置信度传播（affinity propagation）进行图像重排序。给定查询图像，我们提取它的特征并以此查看倒排表，得到一个N维向量，表示它与数据集中每个图像之间的匹配特征数。将这个向量规范化，就得到初始的枢纽值：$\mathbf{w}_0 = (w_{0,1}, w_{0,2}, \ldots, w_{0,N})$，$\sum_n w_{0,n} = 1$。我们在ImageWeb结构$\mathcal{G}$上应用HITS算法[209]，迭代地更新图像的枢纽值和权威值。算法共进行R轮，其中R被称为ImageWeb的**深度**，其直观含义为：所有与查询图像的最短路径长度不超过R的图像都能够被找到。出于时间开销的考虑，我们并不总是等待算法收敛（见第6.3.3节）。图6.16展示了ImageWeb上链接分析算法的一般步骤。

图 6.15　在一个小型的ImageWeb（$N = 1104$）中，两个节点以及它们对应的出边节点（$K = 10$）。对于每个节点，出边对应的图像按照匹配的特征个数（小括号内的数字）进行排序，其中的假正例以红框标示。两个例子中，上面的例子比较简单（只有1个假正例），而下面的例子比较困难（有6个假正例）。

算法4：在ImageWeb上进行检索

1. **输入：**
   ImageWeb结构$\mathcal{G}$，初始权值（枢纽值）$\mathbf{w}_0$，ImageWeb的深度R。
2. **计算第$r+1$轮的权威值：**
   对于$n = 1, 2, \ldots, N$，设权威值$w'_{r+1,n} = 0$；对于每个v_n的入节点v_{i}，令$w'_{r+1,n} \leftarrow w'_{r+1,n} + w_{r,\mathrm{i}}$。将$\mathbf{w}'_{r+1}$规范化，作为所有图像的权威值。
3. **计算第$r+1$轮的枢纽值：**
   对于$n = 1, 2, \ldots, N$，设枢纽值$w_{r+1,n} = 0$；对于每个v_n的出节点v_{o}，令$w_{r+1,n} \leftarrow w_{r+1,n} + w'_{r+1,\mathrm{o}}$。将$\mathbf{w}_{r+1}$规范化，作为所有图像的枢纽值。
4. **迭代：**
   将上述过程重复R次。
5. **输出：**
   最终的枢纽值$\mathbf{w}_R$。

图 6.16　ImageWeb：链接分析（基于HITS[209]）的一般步骤。

6.3.2.4　时间和空间复杂度

我们分析ImageWeb以及相关算法的时间和空间复杂度。在这里我们并不考虑特征抽取、量化、索引的时间开销，以及将大量图像存入倒排表的空间开销。

ImageWeb的时间开销分为两部分：离线建立ImageWeb以及在线查询过程。离线建立任务的主要部分是进行初始检索，其时间开销大部分取决于我们所采用的基准算法。在实际应用中，我们采用标量量化（Scalar Quantization，SQ）[83]生成初始查询结果。假设数据集中共有N张图像，每张图像有M个特征，每个特征在查询过程中被扩展Q次，那么初始查询花费的时间为$O(NMQ\alpha)$，排序花费的时间为$O\left(N^2\log(N)\right)$。这里，α是平均每个倒排表单元索引的特征数量。在实际应用中，有$MQ < N$以及$\alpha \sim \log(NM) < 2\log(N)$，从而建立ImageWeb的时间复杂度不超过$O\left(N^2\log(N)\right)$。此外，向ImageWeb中插入一张图像或者删除一张的时间复杂度是$O(N\log(N))$，在实验中，（见第6.3.4节），N大约是10^6，因此我们需要利用并行算法加速上述过程。考虑到创建ImageWeb在较长的时间内只需要进行一次，这样的时间开销是可以接受的。

ImageWeb的在线查询过程需要花费$O(N\log(N))$时间用以进行初始排序，并且在R轮迭代中进行$2NKR$次浮点运算。根据超参数的不同，时间开销可能从数十毫秒到数十秒不等。选择合适的参数用于查询，这本身也是一个困难的问题。我们将花费一个小节（第6.3.3节），采用折中的观点来讨论这个问题。

ImageWeb的空间开销主要用于存储每张图像的前K名检索结果。存储一张图像结果需要花费8字节（图像ID4字节，图像得分4字节）。根据第6.3.3.2节的结果，我们设定$K = 20$，这样一张图像就只需要花费160字节。据此，在一个1百万张图像的数据集上建立ImageWeb的花费为160M字节，远小于建立倒排表需要的大约4G字节。

6.3.3　参数选择过程的折中思想

本节希望利用折中的思想寻找ImageWeb里的最优参数。为此，我们观察两个现象：准确率与召回率的折中，以及时间复杂度与检索精度的折中。我们发现，折中后的算法不仅能够产生良好的检索精度，而且比基准算法的速度更快。

6.3.3.1　实验设定

我们使用SQ算法[83]作为基准算法。在SQ提供的初始检索结果上，我们构建ImageWeb（第6.3.2.2节）。为了做出公平的比较，我们保留了一些传统方法中通用的实验设置。

- **特征抽取。** 一张图像首先被重置大小，在保证其长宽比不变的情况下，将其长边重置为300个像素。我们使用在DoG检测区域[4]上抽取的SIFT特征[4]。
- **特征量化。** 我们使用标量量化（SQ）[83]将每个128维的SIFT特征量化为256维的0/1编码。
- **索引。** 我们选取256位中的前32位用作倒排表的索引地址，而后224 = 256 – 32位则与图像ID一起存储在倒排表中。出现在超过$N^{1/3}$张图像（N是数据集大小）中的特征作为停用词被筛除。
- **在线检索。** 我们利用SQ的基准检索算法[83]获得初始检索结果并且构建ImageWeb。类似HITS的算法（第6.3.2.3节）被用于检索结果的重排序。其中几个重要参数，将在第6.3.3.2节里一并进行讨论。
- **我们使用检索结果的mAP值作为精度判断标准。**

可调节的参数共有四个：码本扩展阈值（codeword expansion threshold）、海明阈值（Hamming threshold）、ImageWeb的宽度K和深度R。虽然不同的参数对于检索结果影响很大，但为了得到好的结果而将所有可能的参数组合逐一进行测试也是不现实的。我们的目标是利用一个折中的思想来寻找较优的参数组合。在本节接下来的部分，我们将在混合有1百万无关样本的**DupImage**数据集[107]上报告结果。

6.3.3.2　准确率与召回率的折中

首先我们考虑ImageWeb的参数，即宽度K和深度R。显然，一个较大的K可以使ImageWeb为每张图像存储更多相关的样本，而一个较大的R则可以使置信值传播得更远。直觉告诉我们，增加K和R分别有利于提升检索结果的召回率和准确率。正如[216] [75]所示，图像检索系统往往需要在准确率和召回率之间取得平衡。举例来说，降低停用词的阈值（更多较低频词语被停用）将会提升检索的准确率，但同时也会使召回率降低。

我们测试了不同的K和R的组合，并且将所得的mAP值总结在图6.17中。很容易观察到，mAP值并不是始终随着ImageWeb宽度和深度的增加而增长，尽管它们的确对于召回率和准确率的提升有帮助。事实上，当宽度K太大的时候，初始检索中很大一部分的负例就会影响最终结果的精度；而当深度R太大的时候，我们也可能在置信链（affinity chain）中走得太远，从而找到一些已经与查询无关的样本。因此我们得出结论：单独将ImageWeb的宽度和深度最大化，并不会得到最好的检索结果。在图6.17中，最好的结果是折中的体现：宽度$K = 20$，深度$R = 10$。

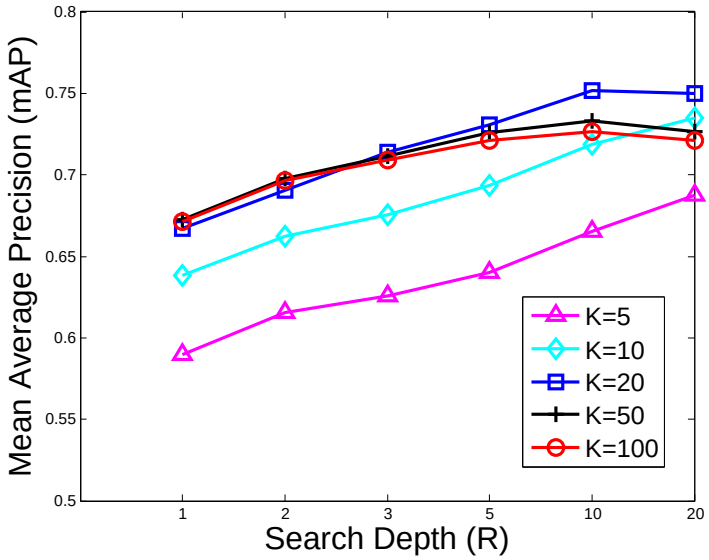


图 6.17　不同的深度和宽度产生的准确率对比。对于每个宽度$R = 1, 2, 3, 5, 10, 20$和深度$K = 5, 10, 20, 50, 100$的组合，我们都报告了相应的结果。实验是在**DupImage**数据集混合1百万无关样本后进行的。

6.3.3.3　初始检索与后处理的折中

接着，我们配置初始检索的参数，即SQ算法中的两个阈值：码本扩展阈值d和海明阈值κ。我们将不同算法产生的检索精度和时间开销列在表6.7中。当$K = R = 0$（不进行后处理）时，算法的精度确实会随着d的增加而显著上升，因为较大的码本扩展能够捕捉更多的匹配特征；然而参数d取较大值的明显缺陷就是造成了初始检索算法的时间开销大幅增加：从$d = 2$到$d = 3$，时间开销增加了近8倍。幸运的是，当采用ImageWeb进行后处理后，初始检索的精度就不再是决定性的了。从表中可以发现，ImageWeb能够将$d = 0$、$\kappa = 16$的初始精度0.143提升至0.752，几乎已经是最好的结果。这说明ImageWeb通过置信传播，可以达到显著提升召回率和准确率的效果。同时，在$R = 10$、$K = 20$下，$d = 0$和$d = 3$产生的结果，在经过后处理后，精度差别非常小（小于0.01）。考虑到$d = 0$只需花费约50毫秒，几乎是$d = 3$的1%，采用$d = 0$是一种非常有效的加速手段。

我们同时为上述实验结果提供一个直观的解释。假设我们固定了在线检索算法的总时间开销，并且希望得到最大的mAP值。此时根据边际效应，无论在初始检索还是在后处理过程上花费绝大部分时间，都是不合理的。通过观察表6.7中两部分时间所占的百分比，就可以得出这样的结论：最有效的检索算法应当让两个过程花费的时间相当。因此，我们选定了一组最佳参数并且在后续实验中使用：$d = 0$、$\kappa = 16$、$K = 20$、$R = 10$。需要注意的是，此时初始检索和后处理分别花

d	κ	K	R	t_1	t_2	t_2/T	mAP	d	κ	K	R	t_1	t_2	t_2/T	mAP
0	16	0	0	50	0	0.00%	0.143	0	24	0	0	54	0	0.00%	0.161
0	16	10	5	50	16	24.24%	0.693	0	24	10	5	54	16	22.86%	0.699
0	16	10	10	50	31	38.27%	0.728	0	24	10	10	54	31	36.47%	0.726
0	16	20	5	50	30	37.50%	0.731	0	24	20	5	54	30	35.71%	0.730
0	16	20	10	50	60	54.55%	0.752	0	24	20	10	54	60	52.63%	0.756
1	16	0	0	80	0	0.00%	0.428	1	24	0	0	86	0	0.00%	0.451
1	16	10	5	80	16	16.67%	0.724	1	24	10	5	86	16	15.69%	0.726
1	16	10	10	80	31	27.93%	0.742	1	24	10	10	86	31	26.50%	0.745
1	16	20	5	80	30	27.27%	0.737	1	24	20	5	86	30	25.86%	0.744
1	16	20	10	80	60	42.86%	0.755	1	24	20	10	86	60	41.10%	0.758
2	16	0	0	460	0	0.00%	0.515	2	24	0	0	477	0	0.00%	0.542
2	16	10	5	460	16	3.36%	0.733	2	24	10	5	477	16	3.25%	0.731
2	16	10	10	460	31	6.31%	0.753	2	24	10	10	477	31	6.10%	0.756
2	16	20	5	460	30	6.12%	0.749	2	24	20	5	477	30	5.92%	0.752
2	16	20	10	460	60	11.54%	0.757	2	24	20	10	477	60	11.17%	0.761
3	16	0	0	4240	0	0.00%	0.537	3	24	0	0	4320	0	0.00%	0.563
3	16	10	5	4240	16	0.38%	0.741	3	24	10	5	4320	16	0.37%	0.744
3	16	10	10	4240	31	0.73%	0.756	3	24	10	10	4320	31	0.71%	0.755
3	16	20	5	4240	30	0.70%	0.756	3	24	20	5	4320	30	0.69%	0.759
3	16	20	10	4240	60	1.40%	0.762	3	24	20	10	4320	60	1.37%	0.764

表 6.7　我们将不同参数产生的精度（mAP值）和时间开销列在表中，以观察初始检索和后处理的折中现象。t_1、t_2和T分别表示初始检索、后处理以及总时间开销（毫秒）。这些时间都是在单个3.0GHz处理器上得到的结果。

费50毫秒和60毫秒，总共时间开销为110毫秒，远低于基准算法（$d = 2$、$\kappa = 24$、$K = 0$、$R = 0$，约480毫秒[83]），但精度却要高得多。

6.3.4　实验部分

我们的算法将和下述几个算法进行对比：

1. **HVVT**[74]是最初的基于词袋模型的检索方法之一，使用层次化的视觉词典树（Hierarchical Visual Vocabulary Tree，HVVT），包含6层，每个节点分支出至多10个子节点。

2. **HE**[75]采用海明嵌入（Hamming Embedding，HE）筛除不正确的量化特征匹配。遵循原文，我们将海明距离的阈值选定为20。

图 6.18　**DupImage**数据集上的样例图像。许多近似重复图像都经过了几何变换或人工修改，与原图差距较大。

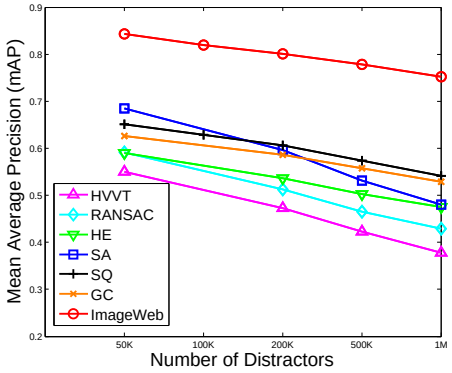


图 6.19　**DupImage**数据集与不同大小的无关样本集合混合后，不同算法产生的结果。

3. **RANSAC**[6]在初始搜索后使用几何验证以筛除负例。它使用随机采样一致性（Random Sampling Consensus）来判断匹配特征集是否来源于仿射变换。

4. **SA**[8]采用软量化（Soft Assignment）将一个特征表示为少数几个视觉单词的加权拟合。从精度和效率折中的角度考虑，我们将其中k维树（kd-tree）的错误界值设为5。

5. **SQ**[83]是我们的基准算法。遵循[83]，我们使用参数$d = 2$和$\kappa = 24$。

6. **GC**[107]采用几何编码（geometric coding）进行局部特征的空间验证。它将特征的空间位置信息压缩存储为一些0/1编码，从而高效地进行匹配判断。

6.3.4.1　DupImage数据集

DupImage数据集[107]包含33个近似重复概念组和1104张图像，包括著名的标志、照片及油画等。图6.18展示了两个典型的概念组。我们还从网络上随机抓取1百万张图像作为无关样本，以测试算法的可扩展性。为了测试算法表现与无关样本数量之间的关系，我们随机选取了4个子集（包含5万、10万、20万和50万张无关样本）。我们测试了ImageWeb算法与上述6个对比算法在这些不同大小的数据集上的结果，同时记录查询的平均mAP值和平均计算时间。

mAP值的对比见图6.19。可以看到，在所有测试的子集上，我们得到的精度都显著地超过了所有对比算法。在最大的数据集上（1百万无关样本），基准算法SQ[83]的mAP值为0.542，而ImageWeb得到0.752，相对提升为38.7%。

图 6.20　**CarLogo-51**数据集上的样例图像。绿框表示标准标志，而红框表示较为困难的样本。

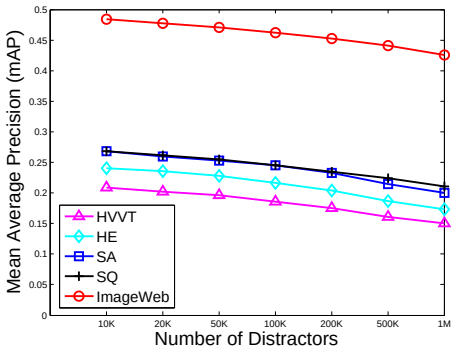


图 6.21　**CarLogo-51**数据集与不同大小的无关样本集合混合后，不同算法产生的结果。

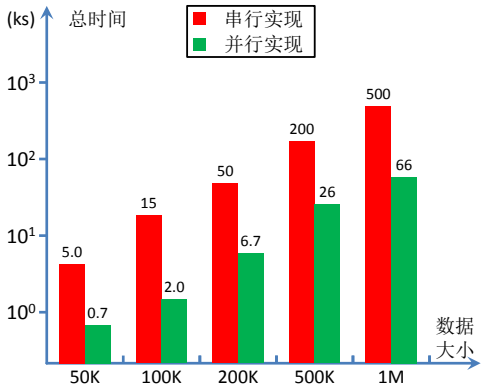


图 6.22　创建ImageWeb的时间开销（千秒）与图像数据集大小的关系。

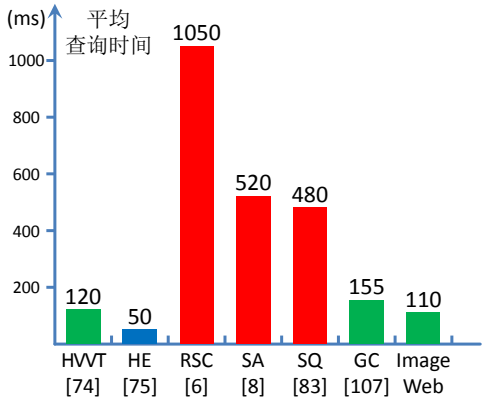


图 6.23　ImageWeb在包含1百万张图像的数据集上的平均查询时间（毫秒）与其他方法的对比。

6.3.4.2　CarLogo-51数据集

我们手工收集了**CarLogo-51**数据集，一个包含51种著名汽车标志的数据库。每种标志都有不少于200张图像样本，整个数据集包含11903张图像。典型的例子如图6.20所示。我们同样将**CarLogo-51**数据集与7种不同大小的无关样本集合（包含1万、2万、5万、10万、20万、50万和100万张无关样本）混合，以测试算法的可扩展性。

mAP值的对比见图6.21。再一次地，在所有测试的子集上，我们得到的精度都显著地超过了所有对比算法。在最大的数据集上（1百万无关样本），基准算法SQ[83]的mAP值为0.211，而ImageWeb得到0.426，相对提升为102%。

6.3.4.3　时间开销

我们算法的离线部分与SQ[83]完全相同，需要大约200毫秒对一张图像进行特

征抽取、量化和索引。这里，我们主要报告两部分时间开销：ImageWeb的创建和在线检索部分。时间和空间复杂度的理论分析可以参看第6.3.2.4节。

图6.22展示了创建ImageWeb的时间开销与数据集大小的关系。值得注意的是，在离线索引完成之后，ImageWeb的创建过程是高度可并行的（每张图像的初始检索是独立的，并行时间加速比几乎呈线性）。考虑到这一过程只需进行一次，这样的时间开销（在1百万图像上少于1CPU天）是完全可以接受的。

图6.23展示了不同方法的在线查询时间开销。注意到我们的算法处理一次查询只需要大约110ms，远比除了HE[75]之外的算法快。相比于基准算法SQ[83]，ImageWeb的查询时间甚至不到1/4。如第6.3.3节的分析，我们充分受益于时间复杂度和检索精度的折中，这使得我们只需要大约60毫秒的后处理时间，就能够将初始检索的时间开销从480毫秒降至约50毫秒。考虑到ImageWeb产生了比基准算法好得多的检索结果，这样的时间复杂度是非常激动人心的。

6.3.5　结论

本节提供了一种图像检索后处理的全新视角。我们的灵感来源于一个显而易见的事实：图像之间的相关关系比特征之间的匹配关系更加稳定可靠。因此，利用一个图像级别的图结构以及在其上进行的置信度传播，我们能够有效地提高基准算法的准确率和召回率。此外，在参数选择方面，我们还利用折中的思想，取得了检索精度和时间复杂度的平衡，有效地降低了时间开销。在检索精度大幅高于基准算法的基础上，ImageWeb算法只需要大约1/4的在线检索时间，这无疑是非常惊人的提升。同时，ImageWeb具有极强的通用性，可以在**任何一个**大规模图像检索问题中获得应用。

6.4　本章小结

本章将视觉词袋模型应用于图像检索任务，并且提出了有效的基于图结构的后处理方法，以提升初始检索结果的质量。我们提出的两个算法，异质图传播（Heterogeneous Graph Propagation，HGP）和图像网络（ImageWeb），具有一定的联系。首先，HGP和ImageWeb的灵感来源是一致的：它们都注意到，使用置信度传播的方式，能够挖掘图像和特征之间更加本质的联系；其次，HGP提出在图像和特征之间传播置信度，而ImageWeb则简化了这一过程，直接在图像之间进行计算，大大简化了算法设计。从时间效率和检索精度上看，ImageWeb都显著地比HGP算法更有优势：这说明将特征匹配作为隐式线索（ImageWeb将特征匹配体现在边的权值上），更有利于产生简单而高效的检索方法。

将来的工作包括将ImageWeb算法应用于真正大规模的网络图像搜索任务。由于网络的图像数据通常具有十亿（billion）量级，当前的算法在存储结构和计算开销上都需要进行改进，以适应大数据的挑战。

第 7 章　统一的图像分类和检索模型

7.1　研究动机

过去的二十年，我们目睹了计算机视觉领域里图像分类和检索算法的繁荣发展。在图像分类方面，物体识别数据集的类别数以及从几十增加到了几万[30]，而且以深度学习为基础的算法[11]被证明能够在这样大规模的物体识别任务中取得非常好的效果。同时，图像检索算法也从一些小规模的实验性程序移植到十亿（billion）级别的实际数据中，越来越多的新问题（如细粒度图像搜索，见第8章）被提出，得到了充分的关注。

图像分类和检索任务都需要处理一定的查询任务。分类任务的目标是判断查询图像所属的类别，为此我们需要一定量的训练样本作为标准；检索任务的目标则是从一个大数据库中找出与查询图像相关的样例，与分类不同的是，检索数据库内的图像往往没有显式的标注。虽然传统的基于视觉词袋模型的方法可以同时处理这两种问题（见第5章和第6章），然而分类和检索算法需要不同的后接模块，以将视觉词袋模型的图像表示转化为分类和检索结果。对于分类，一般需要经过特征的池化和一个额外的训练过程[2][10]；而对于检索，则需要建立索引并进行在线查询和后处理[3][6]。

本章提出使用**在线最近邻估计**（Online Nearest-neighbor Estimation，ONE）算法同时处理图像分类和检索任务。为了达到这一目标，我们考察查询图像和数据库图像之间的相似性，以此作为分类和检索的基础。受到[217]的启发，我们在每张图像上检测多个可能的物体区域，并利用高质量的图像特征描述这些物体乃至整张图像。在在线查询环节，查询图像与某一类（某一张）候选图像的相关性，可以通过特征的平均最近邻距离来计算。根据图7.1展示的结果，抽取更多的物体区域有助于发现更多的视觉线索，从而达到更好的分类和检索结果。我们采用高维的深度特征来描述图像和相应物体（区域）。为了提高效率，我们将深度特征压缩成紧凑的特征编码以进行近似近邻搜索，并且充分利用GPU以加速计算。我们在许多图像分类和检索数据集上进行了实验，这个统一算法能够在合理的时间开销下，达到分类和检索精度的先进水平。

本文的主要贡献可以分为三个方面。首先，我们发现了将图像分类和检索算法合二为一的可能性，即提出ONE算法。其次，ONE算法在多个数据集上达到了先进的分类和检索准确率，说明无需训练的分类算法和基于区域特征的检索算法

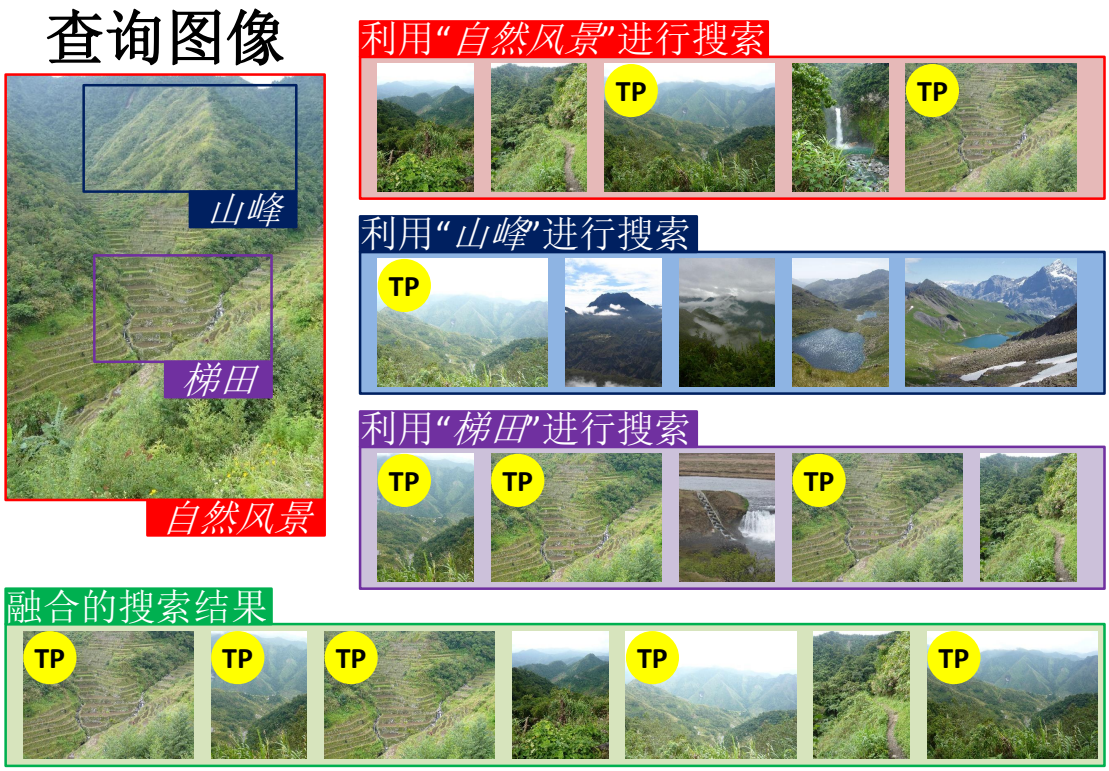


图 7.1　一个图像检索的例子，同时也是我们提出的ONE算法的直观说明。在查询图像上，我们能够通过物体检测找到若干具有语义的物体，它们从不同的角度描述了这张图像。虽然从单个物体出发，无法精确判断查询的意图，但是将分立的检索结果融合起来，就能得到满意的结果。图中标有TP的黄色圆形表示一个真正例（true-positive）。所有的图像都是从**Holiday**数据集上选取的[75]。

也能奏效。最后，我们充分利用了GPU并行，以缓解繁重的在线计算开销。我们的方法可能会对未来许多研究工作提供线索。

与本章相关的出版物为[218]。

7.2　ONE算法

7.2.1　统一的分类和检索模型

我们的目标是设计一个统一的模型，能够同时应用于图像分类和检索任务。因此，我们先进行一个简单的讨论，以确定两类任务（分类和检索）之间的区别和联系。

图像分类和检索算法都需要度量查询图像和候选图像之间的相似度。在分类任务中，每张候选图像（即训练图像）都有一个事先标定的类别：这等价于将所有候选图像分为了若干集合。因此，我们事实上在计算查询图像到每一个候选类（而不是单独一张候选图像）的相似度。依据[217]的结论，图像到类别的距离比图

图 7.2　在同样的查询图像（紫色菱形）下，图像分类和检索过程的区别。红色的方形和蓝色的圆形分别表示对应于书店和图书馆类别的图像。对于每张图像，我们选取其中发现的三个最显著的视觉属性，并且用颜色标注它们的偏向性（红色表示偏向书店，蓝色表示偏向图书馆，绿色表示中立，即不偏向任何一者）。这些候选图像按照它们在特征空间上与查询图像的距离依次编号。双虚线表示两个类之间划分的最优线性分隔线（使用线性SVM计算）。

像到图像的距离更加稳定和可靠。另一方面，检索任务里的候选图像并没有类别标签，因此每张候选图像都作为一个单独的个体进行考虑。我们使用图7.2将两者的区别表示出来。对于图中这张查询图像（属于图书馆类），如果采用最近邻搜索的方式寻找其特征空间中最近的样本，就会找到一张属于书店的图像（标号1），然而这并不是我们希望得到的分类结果，因为这张图像属于异常情况（outlier）；而如果考虑所有训练样本以及由此构建的最优分类平面，就能够得到正确的分类结果。

以上实例说明了分类任务相对于检索任务的天然优势，同时也说明了为什么使用简单的近邻搜索方法往往不能得到良好的分类效果。分类任务能够从图像的类别标签中获得额外的信息从而改进分类结果，然而检索任务无法获得这样的

好处。我们的直接想法是，从每张图像上抽取若干可能的物体区域，从而达到将一张图像扩展为若干张具有相同标签的图像的目的。这样，即使在检索任务中，每张图像也被标注了“类别”信息，从而与若干张其他图像联系在一起。如果每个区域都能够用高质量的描述子表达，我们就有可能通过计算图像到类别的距离[217]，统一地完成分类和检索任务。

7.2.2　ONE算法

本节将上述思想正式地表述出来，成为我们提出的**在线最近邻估计**（Online Nearest-neighbor Estimation，ONE）算法。我们从一个包含N张图像的（分类或者检索）数据集$\mathcal{I}$开始：

$$\mathcal{I} = \{(\mathbf{I}_1, y_1), (\mathbf{I}_2, y_2), \ldots, (\mathbf{I}_N, y_N)\} \tag{7-1}$$

其中，$\mathbf{I}_n$和y_n分别表示第n张图像的数据向量以及类别标签。对于图像分类，$y_n \in \{1, 2, \ldots, C\}$是数据集预先定义的；对于图像检索，我们简单地令$C = N$且$y_n = n$：这表明每张候选图像都属于一个独立的“类别”。

对于每张图像$\mathbf{I}_n$，我们在其上抽取一个可能的物体区域集合$\mathcal{T}_n$：

$$\mathcal{T}_n = \{\mathbf{t}_{n,1}, \mathbf{t}_{n,2}, \ldots, \mathbf{t}_{n,K_n}\} \tag{7-2}$$

此处，$\mathbf{t}_{n,k} = \left(x_{n,k}^{\min}, y_{n,k}^{\min}, W_{n,k}, H_{n,k}, \theta_{n,k}\right)$表示第$n$张图像的第$k$个物体区域，由它的左上角坐标、大小（宽度和高度）、转角$\theta_{n,k} \in [0°, 360°)$（在抽取特征之前的旋转量）唯一确定。$K_n$表示物体区域的总数。$\mathcal{T}_n$的设计将在第7.2.3节详细讨论。将$\mathbf{I}_n$按照$\mathbf{t}_{n,k}$进行切割和旋转，就得到了一个子图像（sub-image）$\mathbf{I}_{n,k}$，在其上能够提取区域特征（描述向量）$\mathbf{f}_{n,k}$. 此处，我们抽取4096维的深度神经网络（deep conv-net）特征[127]，它们是一个深层卷积神经网络的中间输出结果。我们也可以使用其他的特征，如视觉词袋特征（BoVW）[9][7]或者VLAD特征[40]。然而许多事例表明，深度神经网络特征能够在多种不同的数据集上产生良好的效果。以基于最近邻特征搜索的算法为例，在**Holiday**数据集[75]上， VLAD特征能够得到的mAP值为0.526[75]，而深度特征产生的mAP值为0.642[141]。

接着，我们使用[217]提出的朴素贝叶斯最近邻搜索（Naive-Bayes Nearest-Neighbor，NBNN）算法进行图像分类和检索。我们首先定义特征集合$\mathcal{F}_c$，$c = 1, 2, \ldots, C$，表示所有属于第c类的图像中提取的特征的集合：

$$\mathcal{F}_c = \{\mathbf{f}_{n,k} \mid y_n = c \wedge 1 \leqslant k \leqslant K_n\} \tag{7-3}$$

对于一张图像$\mathbf{I}_0$，我们计算它到第c类的距离$\text{dist}(\mathbf{I}_0, c)$，$c \in \{1, 2, \ldots, C\}$，将其定义为从$\mathbf{I}_0$中提取的所有特征到$\mathcal{F}_c$里最近邻的平均距离：

$$\begin{aligned} \text{dist}(\mathbf{I}_0, c) &\doteq \text{dist}(\mathbf{I}_0, \mathcal{F}_c) && (7\text{-}4) \\ &= \frac{1}{K_0}\sum_{k=1}^{K_0}\text{dist}(\mathbf{f}_{0,k}, \mathcal{F}_c) && (7\text{-}5) \\ &= \frac{1}{K_0}\sum_{k=1}^{K_0}\min_{\mathbf{f}\in\mathcal{F}_c}\left\|\mathbf{f}_{0,k} - \mathbf{f}\right\|_2^2 && (7\text{-}6) \end{aligned}$$

当查询图像到所有类的距离都计算完毕后，我们就可以方便地对这些距离进行操作，以达到相应目的：寻找其中距离最小的一类（用于分类），或者将所有距离排序（用于检索）。

7.2.3　感兴趣的物体区域

对ONE算法的描述还剩最后一个部分，那就是对于每张图像$\mathbf{I}_n$定义相应的物体集合$\mathcal{T}_n$。为此，我们可以使用无监督的物体检测子，如物体性检测（Objectness）[156]、选择性搜索（Selective Search）[157] 或者二分规范化梯度（Binarized Normed Gradients，BING）[158]。这些算法能够为每张图像提供一系列包围框（bounding boxes），表示可能的物体区域。通过对每个包围框的置信分数（conficence score）进行排序，我们就可以获得任意数量（K_D个）的物体区域。

另一方面，我们也可以采用一种手工定义的方式来寻找图像的感兴趣区域。为此，我们首先定义一个整数L_O，表示物体区域的层数。随后，我们规定每一层里的物体区域具有同样的大小，并且尽量均匀地分布在图像平面上。假设图像$\mathbf{I}_n$的宽度和高度分别为W_n和H_n。在第l层，有$r_l \times r_l$个物体区域，每个区域的大小都是$\frac{W_n}{s_l} \times \frac{H_n}{s_l}$，其中$r_l$和$s_l$是第$l$层的**物体密度参数**和**物体尺度参数**。遵循均匀分布原则，第(a, b)个物体（$0 \leqslant a, b < r_l$）的左上角就位于$\left(a \cdot \left(W_n - \frac{W_n}{s_l}\right)/(r_l - 1), b \cdot \left(H_n - \frac{H_n}{s_l}\right)/(r_l - 1)\right)$。显然，一个$L_\text{O}$层的模型有$K_{L_\text{O}} = \sum_{l=1}^{L_\text{O}} r_l^2$个物体区域。

无论在自动检测还是手工标注情况下，物体都可以进行旋转。为了简单起见，我们考虑4种最常见的旋转，即$\theta_{n,k} \in \{0°, 90°, 180°, 270°\}$。由于算法的时间和空间开销会随着物体区域的增加而上升，我们可以考虑不旋转所有的区域，而是在自动检测的区域中选择前K_R个（$K_\text{R} \leqslant K_\text{D}$）具有最高置信分数的区域；或者在手动定义的区域中选择前L_R层（$L_\text{R} \leqslant L_\text{O}$）的所有区域。这样的选择性策略将会相应产生$K_\text{D} + 3K_\text{R}$个或者$K_{L_\text{O}} + 3K_{L_\text{R}}$个物体区域。

自动检测和手动定义的区域集合的区别，非常类似于在视觉词袋模型中抽取

特征时，采用检测子和密集采样的区别。我们将在第7.3.2节中说明，两种方法都能够取得令人满意的图像分类和检测效果。

7.2.4　近似最近邻搜索

由于ONE算法需要进行大量的最近邻搜索，线性的暴力搜索可能存在计算代价太高的问题。为此，我们使用*主成分分析*（Principal Component Analysis，PCA）和*乘积量化*（Product Quantization，PQ）[168]算法来降低计算复杂度。4096维的深度特征[13]通过PCA降至D维，随后在PQ算法中被切分为M段。在每段上，我们使用所有的特征训练一个大小为T的码本，并且硬量化每个特征，以进行压缩存储。

值得注意的是，特征在经过PCA降维后，其每一维的能量（包含的信息量）呈递减状态。如果直接利用PQ进行切分，就会导致每一段包含的特征具有不同的能力，从而减弱PQ算法的效果。为此，我们将PCA降维后的向量进行维度置换，使得第m个PQ分段得到降维后的第$(m, M+m, 2M+m, \ldots, D-M+m)$维。在实际操作中，这种简单的改进可以稳定地产生大约2%的分类和检索精度提升。此外，也可以在PCA降维后的向量上利用白化（whitening）方法，达到类似的效果。

假设图像分类数据集里有N张训练图像，或者图像检索数据库里有N张候选图像。每张图像上定义了K个可能的物体位置，而每个物体上抽取了一个4096维的深度特征。对于每一张查询图像，我们需要抽取K个区域特征，从而候选特征向量的总数为KN。暴力的枚举算法的时间复杂度为$O\left(4096K^2N\right)$，空间复杂度为$O(4096KN)$。按照上述近似方法，每个特征向量以PCA降至D维、被PQ切分为M段、每段量化为T个编码之一。根据PCA和PQ的结论（见第2.4.4节），时间复杂度降为$O\left(K^2NM+KDT\right)$；同时，我们只需要存储$KNM\log_2 T$个0/1位，以及DT个浮点数（码本）。由于$M \ll D \ll 4096$以及$T \ll N$，时间和空间复杂度都大大降低了。当这些近似参数确定后，系统的总时间和空间复杂度随着数据集大小N呈线性增长，保证了ONE算法的可扩展性。

在第7.3.4节，我们将从实验角度验证算法的时间和空间复杂度。

7.2.5　GPU加速

尽管经过PCA和PQ算法的近似，在线查询的计算量仍然非常繁重。例如，在包含1百万张图像的数据集上进行一次图像检索，需要大约5×10^{11}次浮点运算（见第7.3.4节）。幸运的是，这些运算中的绝大部分都来源于PQ算法中的线性加

法，我们可以充分利用GPU并行技术来加速这些计算 ①。

值得注意的是，GPU的显示内存（简称为显存）通常比计算机的主内存小得多。例如，NVIDIA公司生产的GeForce-GTX-Titan（当前最强大的GPU型号之一）也只有6G字节的显示内存。为此，我们需要仔细设计算法的参数，使得其存储开销能够适合GPU的限制。此方面的详细实验见第7.3.2节。

7.3　实验部分

7.3.1　数据集和实现细节

对于图像分类，我们测试三个场景分类数据集和三个细粒度物体识别数据集。在场景分类上，我们使用**LandUse-21**数据集[188]（21类航拍图像，每类100张）、**Indoor-67**数据集[189]（67类室内场景，共计15620张图像）以及**SUN-397**数据集[31]（397类室内外场景，共计超过10万张图像）。每类随机选取的训练样本数分别为80、80和50。在细粒度物体识别上，我们采用Oxford **Pet-37**数据集[179]（37种宠物猫狗、7390张图像）、 Oxford **Flower-102**数据集[33]（102种花、8189张图像）以及Caltech-UCSD **Bird-200**-2011数据集[34]（200种鸟类，11788张图像）。每类随机选取的训练样本数分别为100、20和30。 **SUN-397**[31]是当前最大的用于场景分类的数据集之一，可以用于测试算法的可扩展性。对于每个数据集，我们都采用10轮随机的训练和测试数据划分，并且报告平均的分类准确率。

对于图像检索，我们使用两个近似重复图像数据集，即**UKBench**数据集[74]（2550个近似重复组，每组4张图像）和**Holiday**数据集[75]（500个近似重复组，总共1491张图像）。对于**UKBench**数据集，测试标准为N-S分数，即对每个查询图像的返回结果中前4位的正例个数取平均；而对于**Holiday**数据集，则计算标准的mAP（mean Average Precision）值。为了测试算法的可扩展性，我们还将**Holiday**数据集与网络抓取的1百万无关样本混合。

对于手动定义的物体区域，我们设置层数$L_{\mathrm{O}}=5$，并且固定每层的物体密度和尺度参数分别为：$(s_1,s_2,s_3,s_4,s_5)=(1.0,1.2,1.5,2.0,2.5)$和$(r_1,r_2,r_3,r_4,r_5)=(1,2,3,4,5)$。对于自动的物体检测，我们使用选择性搜索算法[157]，并且选择置信分数最高的物体集合。在物体表示方面，我们利用一个事先训练的19层深度卷积神经网络，即**VGG-Net**[127]，在其上提取倒数第2个全连接层（fully-connected

① 图像处理单元（Graphics Processing Unit，GPU）是一个通常用于图形计算和显示的电子元器件。它包含有大量的流处理器（stream processors），可以快速处理大量、简单、规整的代数计算（如矩阵加法、乘法等）。高度并行化的计算结构使得GPU的运算速度通常能够达到CPU的数十倍以上。近年来，GPU也被广泛应用于深度神经网络的训练，并且大大提高了原先的训练效率[11]。

Objects	$L_O = 1$	$L_O = 2$	$L_O = 3$	$L_O = 4$	$L_O = 5$
$L_R = 0$	67.18	73.41	77.63	82.18	85.71
$L_R = 1$	67.40	73.69	77.95	82.42	86.11
$L_R = 2$	–	73.78	78.01	82.47	86.14
$L_R = 3$	–	–	78.03	82.51	86.18
$L_R = 4$	–	–	–	82.53	86.22
$L_R = 5$	–	–	–	–	**86.24**

(a) **Flower-102**数据集上的分类准确率（%）

Objects	$L_O = 1$	$L_O = 2$	$L_O = 3$	$L_O = 4$	$L_O = 5$
$L_R = 0$	0.751	0.772	0.796	0.826	0.847
$L_R = 1$	0.816	0.832	0.837	0.854	0.871
$L_R = 2$	–	0.838	0.848	0.858	0.874
$L_R = 3$	–	–	0.861	0.868	0.880
$L_R = 4$	–	–	–	0.882	0.883
$L_R = 5$	–	–	–	–	**0.887**

(b) **Holiday**数据集上的检索精度（mAP值）

表 7.1　图像分类和检索精度与手动定义物体区域的关系。

layer）上的4096维神经元响应。所有特征在PCA降维之后，都先后使用平方根归一化和ℓ_2范数归一化进行处理。

7.3.2　模型和参数

我们讨论ONE算法的参数，即物体区域集$\mathcal{T}$的定义方式、PCA降维度D、以及PQ的分段数M以及码本大小T，对实验结果的影响。为了进行快速的检验，我们只在两个相对较小的数据集上进行测试，即**Flower-102**数据集和**Holiday**数据集。

我们首先使用精确的最近邻搜索来检验分类和检索算法的精度与物体区域提取算法的关系。表7.1展示了检索和分类精度与手动定义的物体层数的关系；而图7.3则对比了自动检测和手动定义物体产生的精度。我们可以看到，一般来说，抽取更多的物体有助于产生更好的分类和检索精度。同时，自动检测和手动定义的物体产生的精度非常接近，说明物体区域的数量对于提升精度可能更加有用。当提取的物体区域数足够大（$K \geqslant 60$）时，手动定义的区域甚至产生了更好一些的分类和检索结果（如在**Flower-102**数据集上86.24%对86.16%；在**Holiday**数据集上，0.887对0.878）。因此，我们在后续实验中，只使用手动定义的区域，以节省检测算法的计算开销（一张图像大约需要10秒）。

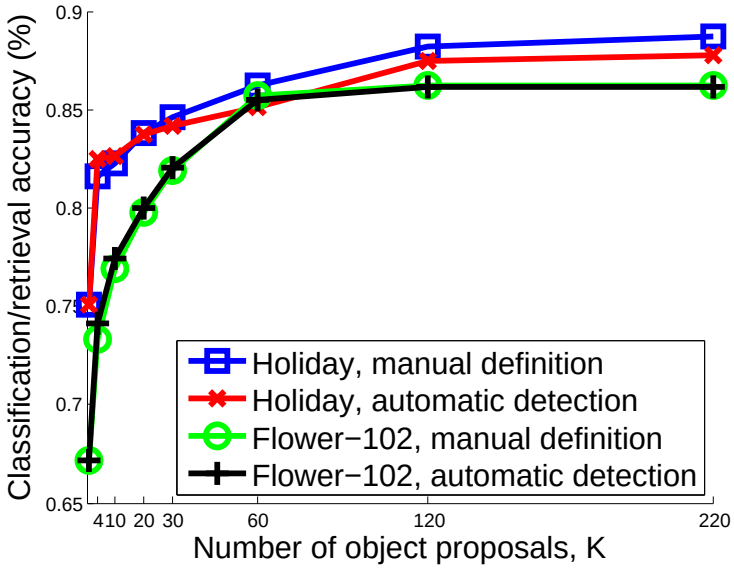


图 7.3　图像分类和检索精度与定义物体区域的方式以及物体区域数量的关系。对于两种方法，我们都首先抽取$K = 220$个物体（对于手动定义，$L_{\mathrm{O}} = L_{\mathrm{R}} = 5$；对于自动检测，$K_{\mathrm{D}} = K_{\mathrm{R}} = 55$），并且从其中随机抽取各种大小的子集，以评测分类和检索结果。

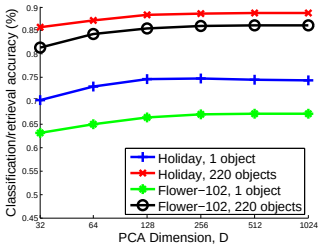


图 7.4　分类和检索精度与PCA降维度D的关系，此处没有使用PQ算法。

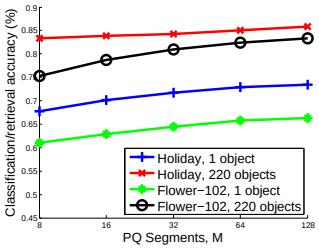


图 7.5　分类和检索精度与PQ分段数M的关系，固定$D = 1024$和$T = 4096$。

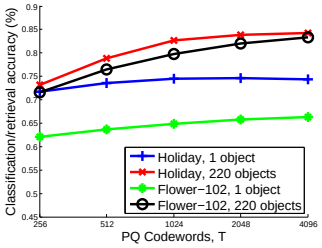


图 7.6　分类和检索精度与PQ码本大小T的关系，固定$D = 1024$和$M = 32$。

在大规模图像搜索任务上，我们考虑使用PCA和PQ算法以加速（第7.2.4节）。图7.4、7.5和7.6 展示了不同参数产生的影响，包括PCA降维度D、PQ分段数M和每段码本大小T。在这些实验中，我们统一使用$K = 220$个手动定义的物体区域，即$L_{\mathrm{O}} = L_{\mathrm{R}} = 5$。经验表明，一个适当的参数集合能够在分类检索精度和计算开销上取得折中，从而达到最好的效果。例如，在PQ算法中，将一个向量分割为128段所产生的精度与分割为64段产生的精度相当，但是却需要几乎两倍的时间和空间开销。从平衡角度考虑，我们选取参数$D = 512$、$M = 32$和$T = 4096$用于后续实验。值得注意的是，在小规模数据集上，利用PQ近似将会显著降低算法的精度：因此在图像数量不超过10^4时，我们只使用PCA而不使用PQ近似。

	LandUse-21	**Indoor-67**	**SUN-397**
Kobayashi[186]	92.8	63.4	46.1
Xie[187]	–	63.48	45.91
Donahue[13]	–	–	40.94
Razavian[141]	–	69.0	–
SVM	**94.52**	68.46	53.00
ONE	93.98	**69.61**	**54.47**
融合	**94.71**	**70.13**	**54.87**

(a) 场景分类准确率（%）

	Pet-37	**Flower-102**	**Bird-200**
Angelova[181]	54.30	80.66	–
Wang[183]	59.29	75.26	–
Murray[85]	56.8	84.6	33.3
Donahue[13]	–	–	58.75
Razavian[141]	–	86.8	61.8
SVM	88.05	85.49	59.66
ONE	**89.50**	**86.24**	**61.54**
融合	**90.03**	**86.82**	**62.02**

(b) 细粒度物体识别准确率（%）

表 7.2　不同数据集上的分类准确率。每一张表的上部和中部分别陈列了不使用和使用深度特征的方法。

此外，我们还需要限制参数的大小，使得所有需要存储的数据（主要是PQ压缩后的编码）能够存储在一个GPU的显示内存中。从第7.2.4节的分析可以知道，PQ需要存储$KNM\log_2 T$个0/1位以及DT个浮点数（后者基本可以忽略），即需要$\frac{1}{8}KNM\log_2 T + 4DT$字节，而一个Titan-GPU的显示内存为大约6G字节。这样，当我们测试一个包含1百万张图像的数据集时，K不能超过120。实际应用中，我们使用$L_{\mathrm{O}} = L_{\mathrm{R}} = 4$（$K = 120$）以处理**Holiday**和**UKBench**数据集的旋转样例。当然，在更多物体区域和更多旋转物体之间，我们也必须做出折中考虑。例如，对于旋转较少的**Oxford Buildings**数据集[6]，使用参数$L_{\mathrm{O}} = 6$和$L_{\mathrm{R}} = 2$（$K = 106$）可能会获得更好的效果。

7.3.3　与现有方法对比

我们将ONE算法的分类和检索精度与现有算法对比。

表7.2中陈列了图像分类的结果。我们可以看到，ONE算法在场景分类和

	Holiday	**UKBench**
融合方法	**0.899**	**3.887**
ONE	**0.887**	**3.873**
BoVW	0.518	3.134
Zheng[134]	0.881	3.873
Razavian[141]	0.843	–
Zheng[220]	0.858	3.85
Deng[212]	0.847	3.75
Zhang[119]	0.846	3.77
Zhang[221]	0.809	3.60

表 7.3　ONE算法的检索精度（mAP）与现有方法的对比。在所有对比算法中，[134]和[141]也使用了**Alex-Net**[11]上抽取的深度特征，而[134]、[212]和[119]采用了后处理算法以提升精度。 ONE算法没有使用任何后处理算法。

细粒度物体识别任务上，都达到了与基于SVM的分类结果相当的准确率。对于细粒度物体识别，我们并没有使用一些复杂的物体部件检测方法，而只是使用规则的手工定义物体区域。由于部件检测在细粒度物体识别上尤其有效，我们报告的精度略低于使用了相应技术的方法，如**Bird-200**数据集上的精度：73.89%[159]和85.4%[219]。当然，我们的算法也可以和这些专门的部件检测算法配合。

由于图像分类任务通常能够从训练过程中获利，我们还比较了将ONE算法与基于SVM的分类器融合后的实验结果。在SVM方面，我们使用LibLINEAR[99]，一个可扩展的SVM模型，并且设置松弛参数为10。在大部分情况下，ONE的分类结果比SVM更好，说明不需训练的分类算法也能够得到很好的结果。当我们将ONE算法和SVM方法产生的结果融合（简单地将它们产生的置信分数相加并且进行重排序），得到的分类结果比单独的算法都好。这说明ONE和SVM提供了某些互补的信息，都能够对分类产生帮助。

接着，我们在表7.3里报告图像检索实验的结果。我们的算法，在不添加后处理模块的情况下，超过了所有的对比方法。与图像分类的情形类似，我们同样将ONE算法与传统的基于BoVW模型的检索结果对比。我们的BoVW模型包括抽取SIFT特征，训练大码本[6]，硬量化，以及使用ℓ_p范数加权[102]。尽管ONE算法得到的结果比基于BoVW算法的结果更好，但我们注意到BoVW算法对于局部特征的描述更加细致。因此，将两种算法融合在一起时，它们也能够互补，从而得到比各自都更好的检索精度。我们强调，在**Holiday**数据集上得到的mAP值0.899以及在**UKBench**数据集上得到的N-S分数3.887，都是尽我们所

	分类 **ONE**	分类 BoVW
特征抽取时间（秒，每张图像）	1.81 (CNN)	1.36 (SIFT)
码本训练时间（小时）	0.39 (PQ)	2.41 (GMM)
码本训练内存（G字节）	0.63	2.50
特征量化时间（秒，每张图像）	0.17 (PQ)	0.23 (FV)
离线训练时间（小时）	–	7.71 (SVM)
离线训练内存（G字节）	–	2.50 (SVM)
在线查询时间（秒，每个查询）	0.08	< 0.01
在线查询内存（G字节）	0.21 (PQ)	0.05
	检索 **ONE**	检索 BoVW
特征抽取时间（秒，每张图像）	1.75 (CNN)	0.83 (SIFT)
码本训练时间（小时）	0.39 (PQ)	6.18 (AKM)
码本训练内存（G字节）	0.63	8.31
特征量化时间（秒，每张图像）	0.17 (PQ)	0.10 (VQ)
离线训练时间（小时）	–	2.85 (IND)
离线训练内存（G字节）	–	4.19 (IND)
在线查询时间（秒，每个查询）	1.17	0.56
在线查询内存（G字节）	5.65 (PQ)	4.19 (IND)

表 7.4　ONE算法和BoVW模型的计算开销的对比。分类的结果在**SUN-397**数据集（包含397类，大约10万张图像）上测试，而检索结果则在**Holiday**数据集与1百万无关图像混合后获得。对于一些缩写词解释如下。 **GMM**：高斯混合模型（Gaussian Mixture Model）；**AKM**：近似*K*-Means算法（Approximate *K*-Means）[6]； **FV**：Fisher向量（Fisher Vectors）[19]； **VQ**：向量量化（Vector Quantization）；**IND**：倒排表（inverted index）。

知的最好结果。当**Holiday**数据集与1百万张无关样本混合后，我们同样得到高达0.758的mAP值（使用了PCA和PQ近似），显著地高于两个对比算法：[134]报告的0.724和[221]报告的0.633。

如此优秀的图像分类和检索结果，归功于物体检测和描述算法的完美配合：ONE算法提供了良好的利用多物体区域进行识别的机制，而深度特征则能够对局部区域提供强有力的描述。无论是将ONE算法替换为简单的最近邻搜索（见表7.1）或者将深度卷积特征替换为其他区域特征（如BoVW特征或者VLAD）都将造成严重的精度下降。

7.3.4　时间和空间开销

这一部分，我们从实验结果分析ONE算法的计算复杂度。理论分析可以参

看第7.2.4节。我们使用第7.3.2节中得到的参数：$D = 512$、$M = 32$和$C = 4096$。表7.4总结了ONE算法在分类和检索算法中的时间和空间开销。

对于图像分类，我们使用实验中最大的数据集——**SUN-397**数据集[31]。遵循作者提供的实验设置，每一次分割中的训练和测试图像的数量都是$N = 397 \times 50 \approx 20\text{K}$。在每张图像上，我们抽取$K = 220$（$L_\text{O} = L_\text{R} = 5$）个可能的物体区域。根据第7.2.4节的分析，对一张图像的分类需要大约4.5×10^8次浮点乘法运算以及大约3.2×10^9次浮点加法运算。在一个Titan-GPU上，上述运算大约需要花费0.1秒，也就是说，在整个测试集上需要花费大约1800秒（0.5小时）。GPU显示内存的耗费大约为200M字节。与基于SVM的分类算法对比，后者在训练和测试过程中需要花费大约8h和2560MB（单个CPU）。ONE算法在时间和空间复杂度上都占有显著的优势。

对于图像检索，我们在**Holiday**数据集[75]与1百万无关样本混合后测试ONE算法。在$N \approx 1\text{M}$和$K = 120$（$L_\text{O} = L_\text{R} = 4$）的条件下，对于一张查询图像的处理需要大约$2.4 \times 10^8$次浮点乘法运算和$4.5 \times 10^{11}$次浮点加法运算。在一个Titan-GPU上，上述运算大约需要花费1.2秒，与传统方法在单个CPU上耗费的时间是相当的。存储45G个0/1位和2M个浮点数需要大约6G字节的存储空间，恰好能够放入一个Titan-GPU的显存里。与现有的利用深度特征的算法[134]相比，ONE算法的时间和空间复杂度都是相当的。

最后值得注意的是，ONE算法需要的实际计算量事实上远远超过对比算法。例如，ONE算法在1百万数据集上搜索一张图像需要进行大约5×10^{11}次浮点运算，而基于BoVW模型的算法往往只需要不超过10^9次[74][83]。为了让ONE算法在合理的时间内产生结果，GPU是一个非常重要的因素。我们注意到，传统的方法通常包含一系列复杂的模块（如倒排表和空间验证），由于使用了大量异步的内存访问（asynchronous memory access）和串行运算（serial operations），从而很难被并行化。在GPU技术，特别是多GPU技术高度发展的今天，我们的算法可能将会受到更多的重视。

7.4　本章小结

本章提出了在线最近邻估计（Online Nearest-neighbor Estimation，ONE）算法，能够统一地处理图像分类和检索问题。我们论证，在高质量的区域特征的帮助下，图像分类和检索任务都能够用简单的NBNN搜索[217]解决。此外，我们充分利用PCA和PQ以及GPU并行技术，以确保我们的算法能够在合理的时间内产生分类和检索结果。虽然ONE算法非常简单，它却非常有效：在许多数据集上，我们

都得到了当前先进的分类和检索结果。

ONE算法的成功能够为未来的研究产生一定的启发。首先我们认识到：图像分类和检索问题的本质是一样的，只需要有效地度量图像之间的相似度，就能够同时解决它们。其次，在图像上抽取更多的区域，通常能够得到更好的分类和检索精度，这说明即使在深度特征的帮助下，图像的表示也远未达到完美。最后，GPU将成为未来计算的趋势，因此设计能够与GPU良好配合的算法将越来越受到重视。

第 8 章 新问题的探索

8.1 研究动机

在这一章里,我们主要讨论两个新的问题，即细粒度图像搜索（Fine-Grained Image Search）及基于视觉内容的网页分析。这两个问题直接而有趣，但是却很少见于现有的研究文献中。研究这样具有挑战性的新问题，既是对之前研究内容的概括和应用，也是对未来工作的一种探索。

与本章相关的出版物为[222] [223]。

8.2 细粒度图像搜索

8.2.1 问题介绍

近年来，基于内容的图像检索系统（Content-Based Image Retrieval，CBIR）快速发展，图像搜索系统的规模已经达到网络级别。基于视觉词袋模型以及倒排表结构，现有的搜索引擎能够在毫秒级别的时间内，在十亿（billion）级别的数据库中寻找相关图像。尽管各种不同的算法都能够提升近似重复（near-duplicate）或者部分重复（partial-duplicate）的图像检索结果，我们却很少发现能够处理细粒度（fine-grained）图像查询的图像搜索引擎。

实际上，在数据库中寻找细粒度匹配的图像样本是常见的用户需求。这里，细粒度匹配的含义是：候选图像与查询图像在语义层面的相似性达到了细粒度的需求。例如，当用户上传一张随手拍摄的，包含鸟或者花的照片时，他自然希望获得一些包含同样种类（生物学）的鸟或者花的图像，以辅助他的理解。当他上传一个汽车标志或者地标建筑图片时，同样也希望能够找到精确匹配的图像。

基于上述想法，我们提出一个新的研究课题，即**细粒度图像搜索**。它与传统的近似重复图像搜索不同，候选内容可能与查询图像有多层的相关性，而我们希望找到那些与查询图像具有精确语义匹配的样本。图8.1展示了一个典型的细粒度查询（*groove billed ani*，一个鸟类物种）的例子。我们希望能够找到其他也包含同一个物种（即*groove billed ani*）的图像样例。如果不能找到精确匹配的图像，那么我们仍然希望能找到语义较为相近的图像，例如*Brewer blackbird*（另一个鸟类物种），而不是其他无关的内容（例如狗、花、建筑物等）。

图 8.1　上部：一个典型的细粒度搜索查询图像，以及三类候选图像（语义匹配、语义近似、不相关）。下部：我们的搜索结果（数据集见第8.2.2.2节）与Google/百度图像搜索引擎的对比，表明特别设计的细粒度搜索算法能够捕捉传统搜索引擎无法感知的用户需求。

值得注意的是，细粒度图像搜索是非常困难的，尤其在一个大数据集中。如图8.1所示，商业搜索引擎如Google和百度都无法有效地处理细粒度的查询请求。这主要是因为这些引擎没有将细粒度语义显式地索引在数据库中，而只是将图像当成一个局部或者全局特征的集合。当然，我们论述的重点在于传统方法“没有做”细粒度图像搜索而不是“不能做”这一问题。我们并没有声称我们提出的算法比Google或者百度更好：在对比算法没有考虑类似问题时，这样的简单比较是不公平的；此外，我们的算法只考虑了有限几个可能包含细粒度概念的物体类（如鸟类、狗、花、建筑物等），但是实际的商业搜索引擎则需要处理网络上可能见到的所有图像，因此没有处理细粒度概念也是可以理解的。这里，我们主要论述一个事实：在传统图像搜索和细粒度图像搜索之间存在着显著的差距，这使得

我们的工作和以往的工作区分开来，成为一个具有创新意义的问题。

在方法上，我们首先正式而具体地描述细粒度图像搜索问题。为此，我们构建一个具有层次化结构的数据集，并且定义一个评分函数用以评价搜索结果的优劣。随后，我们提出一个专用的算法，利用细粒度分类算法的置信分数（confidence scores）来表示图像的细粒度语义属性，并且将这些属性协同索引在原先的倒排表结构中，以确定图像在细粒度属性空间内的位置。最后，我们设计一个高效的在线查询算法用来查看倒排表，并结合细粒度语义属性，得到最后的搜索结果。实验表明，我们的方法能够有效地捕捉细粒度线索，从而得到一些传统方法无法发现的有趣结果。我们已经将数据集公开，期望能够吸引更多的研究者关注这一问题。

我们将这一工作的主要贡献总结如下：

- **新问题。** 我们将细粒度图像搜索规范地描述为一个新的视觉检索（或多媒体信息检索）问题。尽我们所知，这一问题在之前的工作中还很少涉及。我们期望它能够成为一个新的研究方向，吸引更多的研究兴趣，激发新的思想和算法。
- **新数据集。** 我们在若干现有的图像分类和物体检索数据集的基础上，构建了一个新的数据库，专门用于细粒度图像搜索。新数据库中同时包含细粒度相似和近似重复的图像样例，这使得传统方法很难取得良好的结果。我们已经将数据集公开，并且还将继续添加新的类别和图像以扩充之。
- **新框架。** 基于现有的图像分类和物体检索方法，我们提出了一个新的适用于细粒度图像搜索的框架。不论是离线索引模块还是在线查询模块，新算法都与传统方法有着本质的不同。就我们所知，这是在细粒度图像搜索方面的领先尝试，同时也提供了一个基准系统，可供今后的方法进行对比。

8.2.2　问题描述

本节将正式描述**细粒度图像搜索**问题。由于这是一个之前很少研究的新问题，我们首先介绍问题的目标，然后构建一个专用的数据集并且使用层次化的评分函数来衡量搜索结果的优劣。

8.2.2.1　细粒度图像搜索

传统图像搜索问题往往需要寻找一些近似重复或者部分重复的样本，例如地标建筑（**Oxford Buildings**数据集[6]），标志（**FlickrLogo-32**数据集[224]），或者一些重复出现的物体（**UKBench**数据集[74]）。在这些例子中，候选图像或者与查

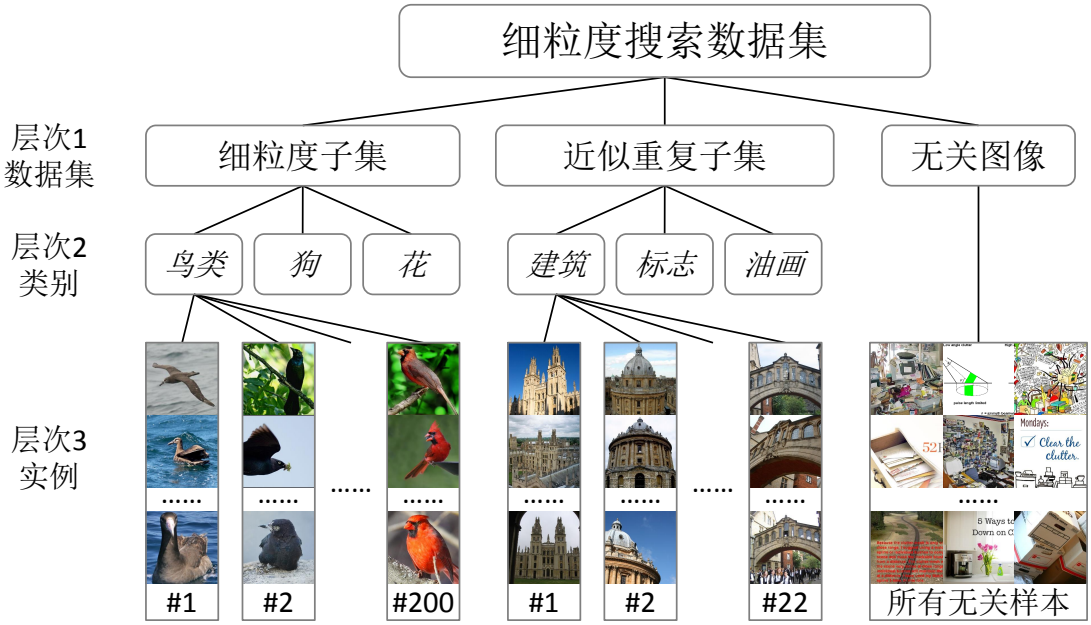


图 8.2　细粒度图像搜索问题的三层数据库结构。

询图像相关，或者不相关，而不可能存在其他情况。在细粒度图像搜索问题中，查询图像有可能包含细粒度语义信息，例如*groove billed ani*（一个鸟类物种）或者*golden retriever*（一种宠物狗）。此时，根据候选图像包含的详细语义属性，它与查询图像之间的相关性就可能分为如下几个不同的层次。

在接下来的部分，我们使用一个三层模型来衡量查询图像和候选图像之间的相关性。两张图像被认为**语义匹配**，如果他们包含精确相同的语义内容，例如相同的鸟类物种、同一个地标建筑物、或者同一品牌的标志；两张图像被认为**语义近似**，如果它们包含基础层面上相同的语义概念，例如*groove billed ani*和*Brewer blackbird*的图像（都是鸟类），或者*Mona Lisa*和*potato eater*的不同样本（都是著名的油画）；除此以外的图像则被认为是**不相关的**，即它们包含的语义信息即使在基础层面上都不相同，例如鸟类、狗、建筑物和油画图像，彼此都是不相关的。

8.2.2.2　数据集

用于细粒度图像搜索的数据库包含三个语义层次，详细情况如图8.2所示。在第一个层次，数据库分为三个部分，即细粒度图像集、近似重复图像集、以及无关图像集。这三个子集的成分如下所示：

1. **细粒度图像集由三个基本语义概念组成：** 鸟类（Caltech-UCSD **Bird-200-2011**数据集[34]）、狗（Stanford **Dog-120**数据集[35]）、以及花（Oxford **Flower-102**数据集[33]）。

数据集类型	数据集名称	概念组数	训练图像数量	检索图像数量
细粒度	*Bird*	200	5994	5794
	Dog	120	12000	8580
	Flower	102	2040	6149
	Subtotal	422	20034	20523
近似重复	*Building*	22	2200	11455
	Logo	52	5200	11903
	Paint	26	2600	3148
	Subtotal	100	10000	26506
无关图像	*Web*	–	3000	1000000
	Total	–	33034	1047029

表 8.1 细粒度图像数据库的组成。

2. **近似重复图像集由三个基本语义概念组成：** 建筑物（**Oxford Buildings**和**Paris Buildings**数据集，总共22个有名称的建筑物[6]）、汽车标志（**CarLogo-51**数据集[110]）、以及著名油画（**FamousPaint-26**数据集）。

3. **无关图像：** 我们从网络上抓取了1百万无关图像加入我们的数据库，以测试算法的可扩展性。

基础层面的视觉概念，例如鸟类、狗、建筑物，被赋予层次索引2；而细粒度或者近似重复的概念，例如*groove billed ani*（一种鸟类）、*golden retriever*（一种狗）、*Oxford all souls*（一个建筑物），则被赋予层次索引3。

在所有细粒度图像数据集中，物体（鸟、狗和花）以它们的生物学物种进行分类。例如，**Bird-200**数据集包括200个鸟类物种，每一物种包含大约60张图像。所有的类别标签，如*groove billed ani*（一个鸟类物种）或者*golden retriever*（一种宠物狗），都可以在**WordNet**[225]，一个大型的本体学（ontology）词典中找到。在所有的近似重复图像数据集中，每个类别对应于一个可重复的物体实例。例如，同一个建筑物在不同角度、不同环境条件下的照片，一个已经注册商标的汽车标志，或者一幅著名油画和它的副本。每个实例的名称（如*Eiffel Tower*、*BMW*标志或者*Mona Lisa*油画）都不具有歧义性。

我们将数据库的一个小部分用于训练。对于细粒度图像数据集，我们使用固定的训练和测试图像划分，并且使用他们天然的生物学类别信息作为图像标签。对于近似重复图像数据集，我们从不是查询图像的集合中，随机挑选每个实例的100个样本用于训练。此外，训练集合还包含3000张无关图像。近似重复图像和

排序	1	2	3	4	5	6
实际分数	3.000	0.000	0.500	0.000	1.161	0.356
理想分数	3.000	1.893	0.500	0.431	0.000	0.000

$$\mathrm{nDCG} = \frac{\mathrm{DCG}}{\mathrm{IDCG}} = \frac{3.000 + 0.500 + 1.161 + 0.356}{3.000 + 1.893 + 0.500 + 0.431} = 0.861$$

图 8.3　一个典型的细粒度图像搜索搜索，包含两个语义匹配、两个语义近似以及两个无关样例。注意到nDCG评分算法能够很好地处理具有多层相似性的情况。

无关图像都被认为**不含有细粒度语义信息**。搜索数据库由除了细粒度训练图像以外的所有图像构成，而查询集合包含所有细粒度测试图像和近似重复查询图像。表8.1概括了我们构建的细粒度图像搜索数据库的组成结构。

当然，还有一些细粒度分类数据集没有被包括进来，例如Oxford **Pet-37**数据集[179]（37个宠物种类，7390张图像）、**Aircraft-100**数据集[36]（100种飞机模型，10000张图像）、以及**Food-101**数据集[226]（101种食物，101000张图像）。未使用的细粒度数据集的样本个数大致相当于已使用的数据集，同时少于无关图像集合。正如第8.2.3.5 节所述，算法的可扩展性使得我们能够驾驭更多的视觉概念。

8.2.2.3　评价方法

我们从定义两张图像**A**和**B** 的相关值开始。一种自然的方式是利用数据库的层次结构：$\mathrm{rel}(\mathbf{A},\mathbf{B}) = \max\{\mathrm{LCA}(\mathbf{A},\mathbf{B}) - 1, 0\}$，其中$\mathrm{LCA}(\cdot,\cdot)$ 是两张图像的最近公共祖先的层次索引值。显然，两张**语义匹配**的图像的相关值为2；**语义近似**的图像的相关值为1，而**无关图像**的相关值为0。

给定查询图像$\mathbf{q}$，我们能够得到包含P 张查询结果图像的集合：$\mathcal{R}_{\mathbf{q}} = \{\mathbf{I}_{\mathbf{q},1}, \mathbf{I}_{\mathbf{q},2}, \ldots, \mathbf{I}_{\mathbf{q},P}\}$。记$\mathrm{rel}_{\mathbf{q},p} = \mathrm{rel}(\mathbf{q}, \mathbf{I}_{\mathbf{q},p})$，$p = 1, 2, \ldots, P$，那么我们就可以计算$\mathcal{R}_{\mathbf{q}}$

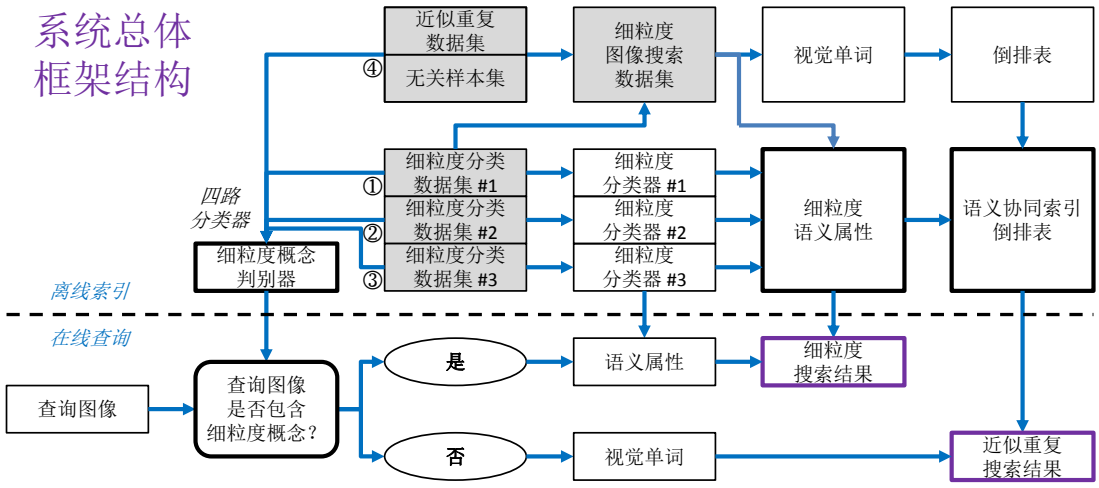


图 8.4　一个完整的细粒度图像搜索框架，包括离线索引和在线查询模块。

的折扣累计收益（Discounted Cumulative Gain，DCG）[227]：

$$\mathrm{DCG}\left(\mathcal{R}_{\mathbf{q}}\right)=\sum_{p=1}^{P}\frac{2^{\mathrm{rel}_{\mathbf{q},p}}-1}{\log_2(p+1)} \tag{8-1}$$

随后，我们将DCG值除以完美查询结果的DCG值（即最大DCG值），得到归一化的nDCG值（位于$(0,1]$内）。图8.3展示了一个典型的评分计算过程。对所有查询图像的nDCG值进行平均，就得到了算法的评分。对于nDCG值的理论分析[228]表明，它非常适用于评价不同排序算法的优劣。

8.2.3　细粒度搜索系统

本节介绍我们用于细粒度图像搜索的算法流程。算法的总体框架如图8.4所示，它主要分为几个模块，将在下面的小节里分别介绍：细粒度识别、大规模特征索引、语义相关协同索引、以及在线查询。

8.2.3.1　细粒度识别

这一模块的目标是训练多个分类器，用于细粒度图像概念的发现和识别。我们使用视觉词袋（Bag-of-Visual-Words，BoVW）模型配合支持向量机（Support Vector Machine，SVM）分类器。

BoVW模型包含三个部分，即局部特征抽取、特征编码以及特征组合（空间池化）。我们从原始图像$\mathbf{I}=\left(a_{ij}\right)_{W\times H}$开始（$W$和$H$分别为图像的宽和高），将其重置大小，在保证图像宽高比不变的情况下，使得长边为600像素，再使用VLFeat代码库[174]抽取128维的灰度RootSIFT特征[118]。对于所有图像，我们保持密集采样

的空间跨度为8，窗口大小为16。 SIFT特征通过PCA降为64维。记降维后的特征集合为：$\mathcal{D}_a = \{\mathbf{d}_{a,1}, \mathbf{d}_{a,2}, \ldots, \mathbf{d}_{a,M_a}\}$，其中$\mathbf{d}_{a,m}$表示第$m$个描述向量，而$M_a$是图像上特征的总数。为了将特征进行量化，我们训练具有256个分量的高斯混合模型（Gaussian Mixture Model，GMM）。用于PCA训练和GMM聚类的特征个数大约为2百万。我们使用改进的Fisher向量（Improved Fisher Vectors，IFV）[7]编码局部特征。在具有256个分量的GMM上，每个区块描述子$\mathbf{d}_{a,m}$将被编码为$2\times256\times64$维特征向量$\mathbf{w}_{a,m}$。记$\mathcal{W}_a$为编码特征集合：$\mathcal{W}_a = \{\mathbf{w}_{a,1}, \mathbf{w}_{a,2}, \ldots, \mathbf{w}_{a,M_a}\}$。我们将使用平均池化将这些特征组合为图像级别的特征：$\mathbf{F}_a = \frac{1}{M}\sum_{m=1}^{M_a}\mathbf{w}_{a,m}$。一个2层的空间金字塔匹配（Spatial Pyramid Matching，SPM）[10]模型将图像进一步切分为2×2子图像，并且将所有5个图像上的池化向量拼接成为一个长向量$\widetilde{\mathbf{F}}_a$，用于图像表示。

为了捕捉颜色特征，我们同样在相同的密集采样区域中，抽取96维的局部颜色统计（Local Color Statistics，LCS）特征[7]。我们采用同样的流程将这些特征编码并组合成为图像级别的表示向量$\widetilde{\mathbf{F}}_b$，并且将$\widetilde{\mathbf{F}}_a$和$\widetilde{\mathbf{F}}_b$拼接为一个长向量$\widetilde{\mathbf{F}}$。

需要注意的是，现今存在一些高级算法[206][90][91]，通过捕捉物体的语义部件信息，从而产生更好的细粒度图像识别结果。然而，这些方法大多基于专门设计的结构模型，因此只能适用于一小部分的视觉概念。我们不使用这些复杂方法，一方面增强了系统的可推广性，另一方面也提高了特征提取的效率。

基于上述图像级别的描述向量，我们训练两大类别的分类模型。第一类模型，称为**细粒度判别器**，是一个四路分类器，通过以下四个训练集得到：鸟类、狗、花以及其他（不包含任何细粒度视觉概念的图像集合）。我们使用这一分类器以判断给定的查询图像是否包含、以及（如果是）包含了哪一种细粒度特征。第二类模型由三个**细粒度分类器**组成，它们分别从每一个细粒度分类数据集上训练得到。将细粒度判别器记为$\mathbb{M}^{\mathrm{J}}$（上标J表示判别，*judgement*），另外三个细粒度分类器记为$\mathbb{M}^{\mathrm{C}}_{\mathrm{B}}$、$\mathbb{M}^{\mathrm{C}}_{\mathrm{D}}$和$\mathbb{M}^{\mathrm{C}}_{\mathrm{F}}$（上标C表示分类，*classification*；下标B、D和F表示鸟类——*birds*、狗——*dogs*和花——*flowers*）。当然，在考虑更多的细粒度概念时，我们还可以训练更多的细粒度分类器。所有的分类模型都使用LibLINEAR[99]（一个可扩展的SVM分类器）和松弛参数10。

8.2.3.2 大规模特征索引

特征索引是大规模图像搜索的关键技术，涉及到BoVW模型和倒排表数据结构。我们同样重置图像的大小，在保证图像宽高比不变的情况下，使得长边为600像素，并且在Hessian Affine区域上[61]提取SIFT特征[4]。记SIFT特征集合为：$\mathcal{D}_c = \{\mathbf{d}_{c,1}, \mathbf{d}_{c,2}, \ldots, \mathbf{d}_{c,M_c}\}$。我们随后使用近似$K$-Means聚类方法[6]训练一个包含1百万单词的大码本，聚类过程中只使用训练图像上的特征。所

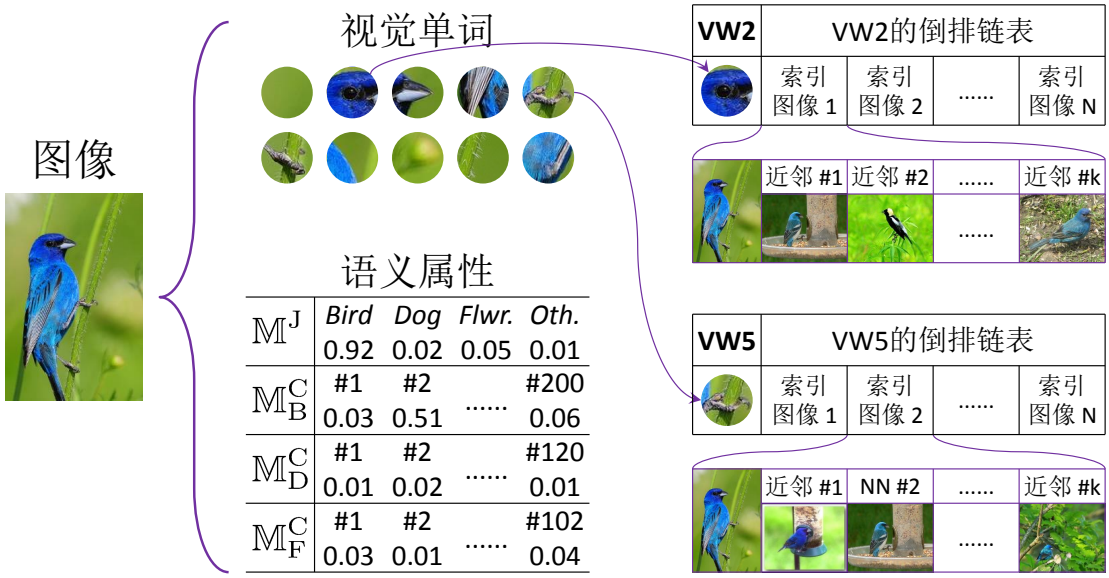


图 8.5　语义相关协同索引后的倒排表结构。对于每张图像，我们都对其提取视觉特征并计算它的细粒度语义属性，再将视觉单词存储在倒排表结构中以供快速查找。在视觉属性空间中的额外最近邻图像也被插入到倒排表结构中，以捕捉更多的语义相似性。

有SIFT特征以硬量化的方式，分配到最近的视觉单词上。记量化后的特征集合为：$\mathcal{W}_c = \{\mathbf{w}_{c,1}, \mathbf{w}_{c,2}, \ldots, \mathbf{w}_{c,M_c}\}$。我们建立倒排表[3]以便后续查询，并且过滤掉其中的停用词（stop words，定义为出现在超过1%图像中的单词）。最后，所有视觉单词通过ℓ_p范数IDF[102]进行加权。

所有的候选图像根据特征匹配数量进行初始查询和排序。随后我们构建ImageWeb（第6.3节），以图结构的方式对图像级别的相关性进行建模，并且采用置信度传播（affinity propagation）方法进行重排序。值得注意的是，还存在着一些能够提供较高mAP值的检索技术[106] [118]，但是这些算法的时间效率通常都比较低（例如需要10秒以上的时间搜索1百万图像数据库），因此无法应用于大规模图像搜索问题中。

8.2.3.3　语义相关协同索引

语义相关协同索引的目标是将图像的细粒度语义属性与倒排表结构相结合，从而使得通常用于捕捉近似重复特征的倒排表结构也能在在线查询环节提供足够的语义信息。我们遵循[221]中的方法，计算图像在细粒度属性空间上的置信分数，并且相应地修改和扩充倒排表结构，使其能够协同索引细粒度语义线索，如图8.5所示。

语义属性，指的是那些能够辅助描述物体的语义概念信息[229]。我们使

用第8.2.3.1 节训练的细粒度分类器 $\mathbb{M}_{\mathrm{B}}^{\mathrm{C}}$、$\mathbb{M}_{\mathrm{D}}^{\mathrm{C}}$ 和$\mathbb{M}_{\mathrm{F}}^{\mathrm{C}}$计算图像的属性分数。例如，$\mathbb{M}_{\mathrm{B}}^{\mathrm{C}}$的输出是一个200维向量$\mathbf{s}_{\mathrm{B}}$，其中的元素$s_{\mathrm{B},k}$表示图像包含第$k$ 种鸟类的置信程度。我们使用softmax函数将评分向量归一化：$\widetilde{s}_{\mathrm{B},k} = \frac{\exp\{s_{\mathrm{B},k}\}}{\sum_k \exp\{s_{\mathrm{B},k}\}}$，从而满足$\sum_k \widetilde{s}_{\mathrm{B},k} = 1$。归一化后的向量$\widetilde{\mathbf{s}}_{\mathrm{B}}$、$\widetilde{\mathbf{s}}_{\mathrm{D}}$ 和$\widetilde{\mathbf{s}}_{\mathrm{F}}$ 被拼接成一个长属性向量$\widetilde{\mathbf{s}}$，其维度为200 + 120 + 102 = 422。

这样，两张图像$\mathbf{I}_a$和$\mathbf{I}_b$ 的视觉属性距离就可以通过它们的属性向量之间的*总方差距离*（Total Variance Distance，TVD）[221]来衡量：

$$\mathrm{TVD}(\mathbf{I}_a, \mathbf{I}_b) = \mathrm{TVD}(\widetilde{\mathbf{s}}_a, \widetilde{\mathbf{s}}_b) = \sum_k \left| \widetilde{s}_{a,k} - \widetilde{s}_{b,k} \right| \tag{8-2}$$

在计算完毕所有图像对的TVD距离后，我们通过两个算法修改倒排表结构[221]，即*语义分隔图像删除*（semantic-isolated image deletion）以及*语义近似图像插入*（semantic-nearest image insertion）。语义分隔图像删除过程枚举每个倒排表中包含不少于3张图像的索引入口，并且检查对应的图像：那些与所有其他图像的TVD值都大于固定阈值ρ 的图像被判定为*语义分隔*，并且从倒排表中删除。这一过程能够有效地降低索引结构的大小，同时不影响在线搜索过程的检索精度。相反地，语义近似图像插入过程则向倒排表内添加具有一定语义相似性的图像。它根据TVD值，寻找每张图像的固定数目的最近邻图像，如果这些图像不在当前视觉单词的索引中，就将其加入图像后的索引结构。从本质上看，这一过程可以看作是基于可信任候选图像的查询扩展。

最后我们强调，仅仅通过三个细粒度数据集（422 个类别）来表示图像的语义属性，是远远不够的。然而，我们的算法在将细粒度分类信息和倒排表结构结合在一起方面，可以作为一个基础可行的尝试。随着更多的细粒度视觉概念被引入，例如*猫*[179]或者*飞机*[36]的图像，我们可以提取更长的视觉特征向量，以编码更加丰富的语义属性信息。

8.2.3.4　在线查询

当遇到查询任务时，我们首先遵循之前的方法，在查询图像上提取所有需要的特征，包括一个图像级别的描述向量$\widetilde{\mathbf{F}}$（第8.2.3.1 节）、一个视觉单词集合$\mathcal{W}_c$（第8.2.3.2 节）、以及一个语义属性向量$\widetilde{\mathbf{s}}$（第8.2.3.3 节）。随后的在线索引流程如图8.4的下部所示。我们使用预先训练的四路分类器$\mathbb{M}^{\mathrm{J}}$判断查询图像是否包含细粒度概念（即*鸟类*、*狗*或者*花*的一种），并且根据判断结果执行下面两种不同的操作。

如果判断结果说明查询图像确实包含某一细粒度概念，例如包含*鸟类*概念，

那么我们就取出相应图像的200维*鸟类*属性向量（由分类器$\mathbb{M}_{\mathrm{B}}^{\mathrm{C}}$产生），将其与数据库内的图像对比。所有图像按照与查询图像的TVD值TVD$\left(\widetilde{\mathbf{s}}_{\mathbf{q},\mathrm{B}},\cdot\right)$进行排序。由于我们能够索引多达1百万图像，逐一计算每个候选的TVD值将会产生巨大的计算开销。此时，我们使用*局部敏感哈希*（Locality-Sensitive Hashing，LSH）方法[167]将图像索引为一些局部敏感的哈希表，并且只在那些与查询图像至少共享一个公共哈希值的候选中进行计算和排序。实际应用中，LSH方法比暴力搜索快得多，而且也能产生满意的查询结果。

如果判断结果表明查询图像并不包含任何细粒度概念，我们就采用语义相关在线查询算法[221]来查看倒排表。此时，搜索过程与一般情况下的近似重复图像检索非常类似，只不过我们充分利用了协同索引的结构来提高检索结果的质量。简单地说，该算法利用语义近似图像插入过程中找到的样本来更新每个候选图像的匹配分数：这一过程等价于将查询图像在属性空间中进行扩展，而扩展的样本就是那些和它最相似的候选图像。当判断器产生错误（图像实际上包含细粒度概念）时，这一模块将尤其有效（参见图8.7的下半部分）。

值得注意的是，我们采用了一个提前决策（early decision）的方法，在查询阶段的开端，就确定了图像是否包含细粒度概念。尽管这样的策略限制了模型的灵活性，使得判断器产生错误后的搜索结果非常不稳定（见图8.7），但它同样带来了一个好处，就是在线查询的时间复杂度显著降低。我们期望将来有更多先进的方法能够改进这一查询过程。

8.2.3.5　可扩展性分析

当前，我们的算法只考虑了很少几个基础层次的类别，即*鸟类*、*狗*和*花*。因此在这里，我们讨论将算法扩展到更多数量的（基础层次和细粒度）视觉概念的可能性。

当基础层次的视觉概念增加时，我们首先需要计算更多的语义属性，并且将属性得分存储在倒排表中。假设我们处理100个基础层次的语义概念，每个基础概念包含大约200个细粒度类别，这就形成了大约20000个类别，规模与**ImageNet**数据集相当[30]。在这种情况下，提取所有的100 + 20000个语义属性将会消耗大量的时间和空间资源，因此我们使用一种近似的方法进行处理。我们首先计算所有100个基础语义属性值，挑选其中具有最高分类置信值的前5名，再对这5个基础类中包含的细粒度类别，计算相应的属性值（这将产生大约5×200个细粒度属性值）。对于其他基础语义概念，我们简单地假设它们包含的所有类的细粒度属性值都相同。

上述方法能够使离线索引过程的时间和空间开销显著降低。在此过程中，唯

一可能的信息丢失发生在真实的基础类不存在于判断结果前5位的情况。根据我们的分类算法在Caltech101数据集[28]（包含101个一般的物体类）上的表现，使用每类30张训练图像（比细粒度搜索数据集的实际情况少得多）就能够达到90%以上的5选准确率。也就是说，使用上述近似方法，时间和空间复杂度都能够降低到大约5%，而相对的信息丢失则少于10%。

当数据集规模扩展后，对于在线查询环节的主要影响来源于在语义属性空间中寻找近似最近邻样本的开销。由于扩展的属性空间可能包含上万维，即使是LSH算法也需要十分繁重的计算量。为此，我们采用另外一种近似策略：独立地在每个基础类别的语义属性空间中进行检索。我们同样寻找前5名的基础类别，在其中每个基础类别里，利用相应的语义属性寻找前1000名最近邻样本。最后，我们将找回的5个列表合并起来，而图像的排序依据就简化为候选图像在这5个属性空间内的总方差距离。这样，我们就能够避免计算大量欠相关的基础类语义属性，无谓地消耗计算资源。

我们将在第8.2.4.4 节提供一个基于实验的时间和空间开销分析。同时，我们也注意到许多其他的方法能够用于处理类似的问题，例如使用层次化的分类结构和一个自顶向下的方法对于图像进行从粗到细的分类。在这里，我们仅仅提供一种可能的解决方案，并且也期待将来在此方面能有更多的研究进展。

8.2.3.6　可推广性分析

在细粒度搜索问题中，三层的相关性结构或许还不足以描述两张图像的相似性。一个更加自然的方法是使用自然的本体结构[225]以计算多层的相关性。例如，[230]使用层次化的语义索引来提升图像检索的效果；或如[231]使用深度学习的方法进行细粒度相似性学习。以上两种方法启发我们将细粒度图像概念的层数增加，以便更好地定义细粒度搜索问题。

将我们提出的两层模型进行修改，就能适应新的问题设定。我们保留基础级别的概念判断模块，并且在细粒度分类过程中使用层次化的损失函数。在这里，层次化损失函数的含义是每个测试图像的分类结果不再只有正确（得1分）和错误（得0 分）两种情况，而允许我们根据测试结果和真实标签的相关程度，赋予一个$[0, 1]$之间的得分（相关系数）。这种方法被广泛地应用于大规模图像分类任务中，用以对基础层次的分类错误（如将*鸟类*图像判断为*建筑物*）施以更重的惩罚。需要注意的是，损失函数的修正并需要我们对现有的算法流程进行任何修改。

	鸟	狗	花	其他	准确率
鸟	5574	102	12	106	96.20%
狗	45	8407	12	116	97.98%
花	83	102	5941	23	96.62%
其他	1	0	5	416	98.59%

表 8.2　细粒度概念判断的混淆矩阵。为了让结果更清楚，102张鸟类图像被识别为狗。

	鸟	狗	花
我们报告的准确率	42.56%	35.29%	78.72%
最好的准确率（不用部件模型）	≈ 44%	≈ 37%	≈ 80%
最好的准确率（使用部件模型）	≈ 56%	≈ 48%	≈ 84%

表 8.3　细粒度物体识别的准确率。

8.2.4　实验部分

8.2.4.1　识别和检索精度

首先，我们报告细粒度概念判断和物体识别（分类任务）的准确率。结果分别陈列在表8.2和表8.3中。可以发现，我们的算法产生了令人满意的细粒度判断结果： 96.20%的鸟类、97.98%的狗、96.62%的花 以及98.59%的其他（不包含任何细粒度概念）查询图像都能够被正确地判断。同时，我们产生的细粒度物体识别准确率接近先进水平（在不使用复杂的部件检测模型的算法中[89][90][91]）。这就保证了在大多数情况下，我们的算法都能够准确地感知用户的意图，从而在后续执行正确的流程。

接着，表8.4报告了近似重复数据集与1百万无关图像混合后，物体检索的精度。我们的物体检索模块，在完成检索时间少于100毫秒的算法中，再次达到了先进水平。由于一些复杂的方法[106][118] 需要较长的查询时间，考虑到系统效率，我们没有采用这些方法。

以上分立模块的良好表现帮助综合框架产生了令人满意的细粒度搜索结果。

	建筑物	标志	油画
我们报告的mAP值	0.7631	0.4842	0.5857
最好的mAP值（不超过100毫秒）	≈ 0.78	≈ 0.49	≈ 0.60
最好的mAP值	≈ 0.87	≈ 0.53	≈ 0.66

表 8.4　使用近似重复查询图像（建筑物、标志和油画）（并包含1百万无关图像）时，得到的平均精度（mAP值）。

	Model-SA	**Model-VW**	**Model-COM**
Bird-200	**0.6741**	0.3215	0.6169
Dog-120	**0.7102**	0.3692	0.6727
Flower-102	**0.7961**	0.4250	0.7302
Building	0.5601	**0.9205**	0.8887
Logo	0.3406	**0.5940**	0.5432
Painting	0.4091	**0.6703**	0.6610
Fine-Grained	**0.7268**	0.3719	0.6733
Near-Duplicate	0.4366	**0.7283**	0.6976
Overall	0.5817	0.5501	**0.6855**

表 8.5　三个不同模型的细粒度图像搜索精度对比。

这里需要说明，我们无意将分类和检索模块与当前先进水平的算法进行对比。通常来说，那些复杂的算法或者使用了特殊的线索或者技巧以致难以推广，或者计算复杂度太高，无法应用于大规模搜索问题。

8.2.4.2　细粒度搜索结果

我们使用第8.2.2.3 节定义的方法来评价细粒度图像搜索的精度。为了对比，我们共测试三个不同的模型：分别使用语义属性、视觉单词，以及同时结合了二者的模型（**Model-COM**，参见第8.2.3.4 节）。语义属性模型（**Model-SA**）只使用依据422 个细粒度类别定义的语义属性$\widetilde{\mathbf{s}}$，并且将所有的候选图像根据它们到查询图像的总方差距离（Total Variance Distance，TVD）进行排序。视觉单词模型（**Model-VW**）只使用量化的特征集合$\mathcal{W}_c$来查看倒排表，并且利用经过ℓ_p范数IDF加权[102]的特征匹配数量对候选图像进行排序。这两种方法等价于跳过细粒度语义判断环节，分别默认图像包含或者不包含细粒度语义概念。

表8.5总结了细粒度图像搜索的结果。我们能够看到，在单独的查询集合上，无论是细粒度查询还是近似重复查询，**Model-COM**都无法在三种方法中取得最好的结果。图8.6提供了两个例子，展现了不同类型的查询图像的区别。当查询图像包含细粒度概念时，**Model-SA**更好地捕捉了图像的全局特征；而当查询图像包含近似重复样例时，**Model-VW**则产生了更高的准确率。当两种查询的准确率进行平均后，则是**Model-COM** 取得了最好的结果：这更符合实际情况，即两种查询样例都有可能出现。我们的实验还表明了另一个重要结论：分析用户意图可能比在单独数据集上产生高精度更加重要。

我们再一次强调，我们的目标是提出新问题（细粒度图像搜索），并且建立新

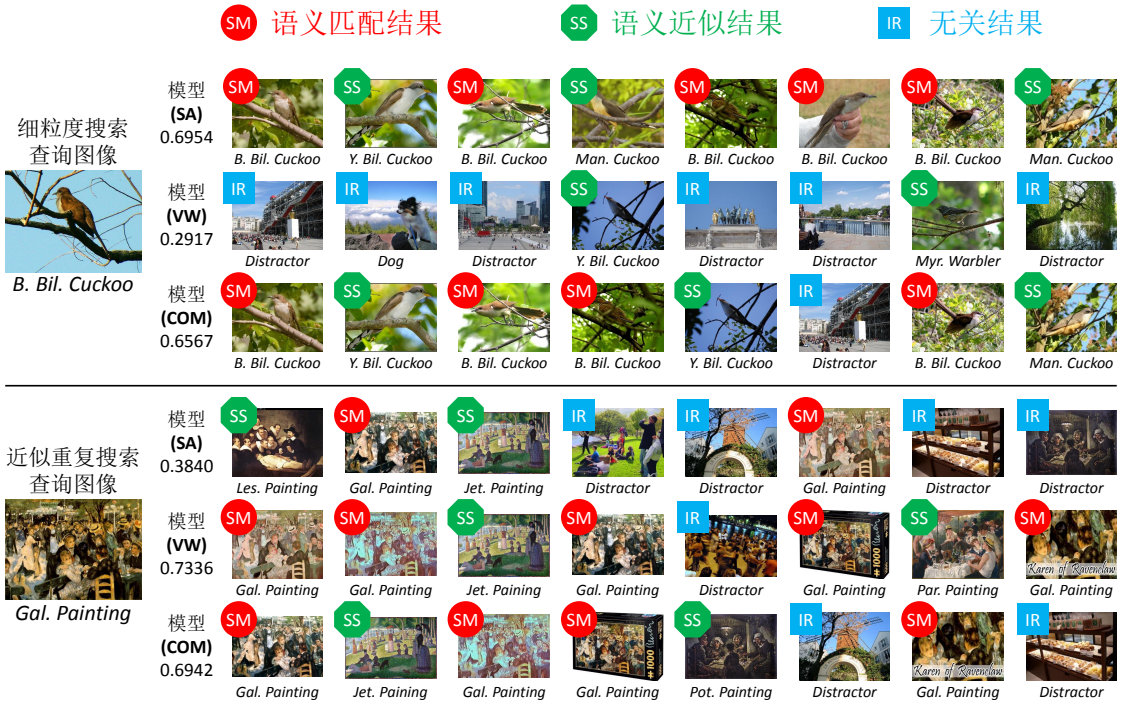


图 8.6　在两张查询图像上，三个不同模型的搜索结果。第一张查询图像包含一个细粒度概念（*black billed cuckoo*，一个鸟类物种），而第二张图像则包含一个近似重复的样例（油画）。每一个模型下方的数字表明了相应方法得到的nDCG分数。

的基准方法。尽管由于我们并没有采用复杂的图像分类和检索算法，导致整体框架的精度未达到很高的水准，然而我们的算法是高度模块化的，因此它能够很方便地与一些未来发展的更有效的算法相结合。我们相信，在使用更加强力的分类和检索算法后，**Model-SA**和**Model-VW**的精度都能够相应地得到提升，从而融合产生的模型**Model-COM**也能够取得更好的效果。

8.2.4.3　时间和空间开销

我们报告在线查询模块在1百万数据集上的时间和空间开销。

我们的算法需要在系统中存储两部分的信息（见图8.5）。第一部分是倒排表中视觉单词对应的图像ID信息：每个图像-单词对需要花费4个字节，平均每张图像包含的特征数量大约是1000个，为此单张图像需要大约4K字节。根据[221]，在语义相关协同索引后，索引需要的空间大约会增加一半。第二部分是根据细粒度物体识别结果计算的语义属性：我们需要存储$4+200+120+102=426$个浮点数，即大约2K字节。因此，每张图像的空间开销为大约$(4\times 150\%+2)\,\mathrm{K}=8\mathrm{K}$字节，整个1百万数据库就需要大约8G字节。另外，在1百万张图像数据库上创建ImageWeb结构[110]，需要的额外开销大约为160M字节。

在线查询环节的时间开销由三个部分组成：特征抽取和量化、细粒度概念判断以及图像搜索。其中，图像搜索有两种可能的流程，利用语义属性进行细粒度查找或者利用视觉单词查看倒排表，这取决于细粒度概念判断的结果。在实际测试中，在查询图像上提取SIFT特征需要1000毫秒（密集采样需要650毫秒，而兴趣点检测和特征提取需要350毫秒），量化特征需要400毫秒（300毫秒用于计算Fisher向量，100毫秒用于硬量化），细粒度概念判断需要50毫秒。如果查询图像被认定包含细粒度概念，那么在接下来的流程中，细粒度物体识别需要100毫秒，而LSH查询需要大约400毫秒。否则，近似重复的图像搜索（利用了语义相关协同索引结构）需要大约550毫秒。最后，使用ImageWeb对搜索结果进行后处理需要大约100毫秒。总体来说，单个查询的时间开销大约为2秒。上述所有的时间开销都是最坏情况（即所有查询图像中记录的最大值）。算法运行于单个3.0GHz的CPU上。

8.2.4.4　可扩展性分析

与第8.2.3.5 节的分析一样，我们仍然假设在基础层次上有100个大类，每类包含大约200个细粒度视觉概念。对于每张图像，我们首先需要存储相关性前5名的基础级别对应的识别结果（ID和相应的分类分数），并计算5×200个细粒度语义属性值。于是，总共需要存储$5 \times 2 + 200 \times 5 = 1010$个浮点数，大约4K字节。如果使用同样数量（1000个）的局部描述子，每张图像的内存开销就是$(4 \times 150\% + 4)\,\mathrm{K} = 10\mathrm{K}$ 字节。实用的图像搜索引擎，如Google和百度，通常需要处理超过10亿张图像，从而需要大约10T字节的存储空间。

在线查询过程的时间开销随着基础类别的数量和索引图像的数量增加而增加。如果查询图像被判定包含某种细粒度语义概念，LSH 算法就需要在5个独立的特征空间（每个大约200维）上计算，这需要花费800毫秒的时间（在422维属性空间中的计算花费大约400毫秒）。将5个查询结果合并起来，需要花费大约100毫秒。如果查询图像被判定为包含近似重复样例，搜索的时间复杂度将随着数据库大小呈次线性增长[110][102]。无论查询图像的细粒度判断结果如何，用于单个查询的时间复杂度都不会超过2.5秒。

也就是说，当基础类别的数量从3增加到100、细粒度类别的总数从422增加到20000、索引图像数量从1百万增加到10亿时，估计的时间（单张图像查询）和空间（单张图像存储）复杂度仅仅增加了25.0%。这样就能够得出结论：我们所提出的算法能够在合理的近似下，方便有效地扩展到超大规模图像搜索问题中。

图 8.7　典型的查询和搜索结果。虚线上方的部分表示被正确判断是否包含细粒度概念的查询图像，而下方的两张鸟类图像、一张标志图像和一张油画图像分别被错误地预测为：不包含细粒度概念、花、鸟类以及花。图像下方的文本标示了它的真实分类标签。我们可以看到，当用户意图（是否包含细粒度概念）被准确判断时，系统产生的搜索结果一般是令人满意的。此外，当细粒度查询被误判为近似重复查询时，语义相似协同检索算法仍然能够产生比较好的检索结果；然而当近似重复查询被误判为细粒度查询时，检索结果就会变得非常不稳定。

8.2.4.5　样例结果

图8.7展示了一些系统搜索的样例结果。我们可以看到，算法在正确判断（是否细粒度概念）的查询样例上工作得很好，说明了解用户需求的重要性。同时，图8.7也呈现了四个错误判断的样例，包括两个细粒度查询样例和两个近似重复查询样例。对于前两个例子（细粒度查询），我们在语义近似协同索引的帮助下仍然能够得到可接受的搜索结果；然而对于后两个例子（近似重复查询），由于我们忽略了关键的局部特征，使得搜索结果变得极不稳定。这再次说明了一点：基础级别的分类准确率比细粒度级别的分类准确率更加重要（见第8.2.4.2节）。

8.2.5　结论

我们提出了**细粒度图像搜索**问题，一个非常有趣但是却很少涉及的研究课题。我们论述：用户通常希望寻找查询图像的细粒度相似样本，而不仅仅是近似重复的样例。为此，我们正式地定义了细粒度图像搜索问题，构建了一个新的数据库，并且采用了一种全新的评价方法。我们还提出了一个基准框架，将高级语义特征与低级视觉单词结合在倒排表结构中，从而设计出一个高效的搜索引擎，能够在大规模数据集上产生令人满意的细粒度图像搜索结果。由于我们的框架是高度模块化的，它能够与多种不同的图像分类和检索方法配合，在模块更新升级后也能够相应地获得效果的提升。同时，算法的可扩展性也使得我们能够很方便地将它迁移到实际的商业搜索引擎中。

我们已经将数据库发布出来，并且将逐渐添加新的细粒度概念和更多的图像，例如**ImageNet**数据集[30]里的概念。第8.2.3.5 节和第8.2.3.6 节讨论的方法能够有效地帮助我们将现有系统进行扩展，从而设计出商业规模的细粒度搜索引擎。我们希望这一研究课题能够揭示一个新的研究领域，并且催生新的思想和新的应用。

8.3　基于视觉内容的网页质量分析

8.3.1　问题介绍

人们通常对于网页的流行度和美观度非常感兴趣。在网页设计与制作的过程中，我们希望训练一个计算机视觉模型，预测网页的流行度和美观度。这里，*流行度*（popularity）指网页能够吸引更多用户访问的程度，我们可以咨询专业的流量统计网站（如Alexa[①]）以获得当前的网站排名；而*美观度*（beautifulness）则指代网页的美学（aesthetics）属性，我们通常需要进行用户调研以获得相关的标注。

① http://www.alexa.com/topsites

图 8.8　四组网页和它们对应的Alexa排名（基于网络流量分析）。从上到下，左边网页的排名比右边网页的排名：高一些、低一些、高很多、低很多。然而在所有的情况下，受试者都更喜欢左边的网页（认为它更美观），说明网络流量分析和用户调研产生的美观度结果可能会有很大的分歧。

在本节中我们将探索下面的问题：在网页的视觉内容和它的流行度/美观度之间，是否存在显著的关系？换句话说，是否有可能设计出一种基于计算机视觉的算法，使得我们能够判断哪个或者哪些网页更具有流行度/美观度？

上述问题可以被归类为设计推荐系统（recommendation systems），即从大量的候选者中，寻找一个高质量的网页子集。这一问题与一些研究领域紧密相关，包括网页内容分析、图像表示、美学分析、用户意图理解，等等。为此，传统的方法通常参考网页的源代码（如HTML 和CSS）[232]，并且从人机交互（human-computer interaction）[233] 和人类工程学（ergonomics）[234]实验中寻找相关的线索；或是建立一个基于超链接（hyperlink）的图结构，在其上执行扩散算法[210] [209]。我们指出，网络用户往往会根据网页的视觉信息，即直接呈现在屏

幕上的内容，来做出访问决定。然而我们几乎无法找到能够基于网页的视觉属性（如排版、配色、字体，等等）来判断它的质量的研究工作。

针对上述富有挑战性的问题，我们提出先驱性的研究方法。我们首先收集一个数据集，包含大约2万张在Alexa上排名靠前的网站的首页截屏（在相同的系统配置下进行）。所有的网页在接下来的研究过程中都被视为纯图像，即网页源代码等文本信息都不再用于分析。随后，我们从数据库中随机选取大量的网页对，并且邀请10名志愿者，根据他们个人的喜好，在每一个网页对中标注更美观的一张。图8.8展示了一些标注的结果，从中我们可以发现：基于网络流量分析的流行度与基于用户调研的美观度之间存在很大的差别。

方法上，我们将预测网页质量（流行度或美观度）建模为一个二分类问题，其中每一个网页对的真实标签来源于Alexa网络流量统计（流行度）或者用户调研（美观度）。我们提倡使用网页截屏中的纯视觉特征来进行分析，而不使用边缘信息，包括文本、标签、链接信息等。我们采用现有的图像分类技术，包括一些基于特征的分类模型（如视觉词袋模型[19]和全局图像特征[235][138]）以及一个专门设计的具有比较神经元的卷积神经网络。实验结果表明，尽管网页的流行度和美观度之间的相关性很弱，但是基于视觉的识别算法仍然可以有效地预测流行度或美观度更高的网页。我们的预测结果能够被迁移到不同的商业应用中，包括网站流行度趋势预测，以及设计个性化的网页推荐系统等。

本节的贡献可以被归纳为三个方面：首先，我们提出了一个有趣却少有研究的课题，即基于视觉信息的网页质量分析；其次，我们利用现有的图像分类技术，建模并探索了这个问题，包括一个专门设计的神经网络模型，并且得到了初步的实验结果；最后，基于网页质量预测算法，我们展示了两个新奇的商业应用。我们将会发布相关的数据库，期望有更多的研究者关注这一问题。

8.3.2　网页质量分析的相关工作

当前，Google已经索引了超过30万亿份网络文档①，而全球网络用户的数量也在2014年超过了30亿。互联网的爆炸性增长使得网页的质量分析和判断变得越来越重要。网页内容的分析或者网络数据挖掘是一个非常有趣的研究领域，它与多个研究课题紧密相关，包括网页排序、图像美学分析以及网络流量分析等。

网页排序算法可以追溯到网络搜索引擎的发展之初，它将排名引入了巨大的互联网。流行的算法，如PageRank[210]和HITS[209]，利用随机游走理论模拟用户的行为，计算用户访问到每一个网页的概率。而当下，更多的因素被列入了排序

① http://en.wikipedia.org/wiki/Google_Search

算法的考虑范围，例如时效性、地域性以及赞助商广告等。新的基于点击模型（click model）的方法也被广泛应用于网页的重排序[236][237]；基于视觉的技术例如图像搜索[24]也能够提供有效的线索。同时，分析用户的意图使得他们能够在访问过程中获得乐趣，也是非常重要的课题[238]。

美学（aesthetics）是心理学的一个分支，主要考虑艺术、美和鉴赏（taste）的本质，以及美的创立与欣赏。众所周知，用户更愿意访问一些具有良好外观的网站。分析网页的美学属性与人类工程学（ergonomics）息息相关，需要考虑不同因素，包括可用性（usability）[234]、交互性（interaction）[233]，以及感官特征（impressive features）（如颜色[239]和物体自然特性（nature）[240]）。由于美学的判断具有极强的主观性[241]，用户调研通常将作为一种重要的数据标注手段[242]。

流行的网页能够吸引更多的用户。特定网站的网页流量分析可以通过时间戳数据（timestamp data）获得[243]。 Alexa网站提供每日更新的全球访问量前1百万的网站名单。

在[244][245] 中，基于视觉的算法已经被用于普通照片的流行度和美观度分析。而我们将讨论用于网页质量判断的视觉算法。

8.3.3　问题设定

8.3.3.1　数据收集

我们下载了Alexa上连续30天的排名数据（2014年8月25日–9月23日，全球访问量前1百万名的网站名单）。我们选择那些出现在全部30个列表中的网站，并且计算它们的平均排名。做平均的原因是每天更新的排名数据不稳定性太强（见第Section 8.3.5.1 节）。其中，我们访问前20000名网站，将其首页的第一屏截图，保存为PNG格式图片文件（在Windows 8.1系统下，使用Google Chrome 38.0版本，屏幕分辨率为1920×1080）。选择排名靠前的网站的一个明显的优点在于，它们的质量通常很高：在前20000名网站中，几乎没有垃圾网站（依靠重复度极高的网页骗取点击率）或者知名网站的假冒副本。这样，用户调研（第8.3.3.3 节）就能够在一个相对清洁的环境下进行。

我们手工筛除一些不合法的截图，即出现下列任一情况的网页：

1. 网页访问失败，通常由网络连接错误造成；
2. 网页包含禁止的内容，如恐怖主义或色情内容；
3. 网页没有正常显示，例如网站正在例行维护，或者网页中的一大部分被一个浮动广告窗口占据；

4. 网页内容与某个更高排名的样例发生重复：这通常发生在一些国际公司（如Google）的不同域名后缀之间。

在手工筛除后，总计$N = 14910$张网页图像被保存下来，构成了一个数据集：$\mathcal{W} = \{(\mathbf{I}_1, r_1), (\mathbf{I}_2, r_2), \ldots, (\mathbf{I}_N, r_N)\}$。这里，$\mathbf{I}_n$和$r_n$分别表示第$n$个网页的图像数据（$1920 \times 1080 \times 3$矩阵）和Alexa排名（$1 \leqslant r_1 < r_2 < \ldots < r_N \leqslant 20000$）。

8.3.3.2　数据分组

我们构建$T = 20$个小组（groups），记为：$\{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_T\}$。每一小组$\mathcal{G}_t$，$t = 1, 2, \ldots, T$，包含$M = 1000$个随机选取的网页对，即：$\mathcal{G}_t = \{\mathbf{p}_{t,1}, \mathbf{p}_{t,2}, \ldots, \mathbf{p}_{t,M}\}$，其中对于$m = 1, 2, \ldots, M$，有：$\mathbf{p}_{t,m} = (a_{t,m}, b_{t,m})$且$1 \leqslant a_{t,m}, b_{t,m} \leqslant N$。我们定义网页对$\mathbf{p}_{t,m}$的**排名差**为两个网页的Alexa排名之差的绝对值：$g_{t,m} = \left| r_{a_{t,m}} - r_{b_{t,m}} \right|$。在小组$\mathcal{G}_t$中，网页对的排名差受到限制：$1000\,(t-1) < g_{t,m} \leqslant 1000t$。这就意味着$\mathcal{G}_1$里的所有网页对都有非常接近的Alexa排名，而$\mathcal{G}_{20}$里的网页对都有很大的排名差。由此，我们可以观察算法预测精度与排名差之间的关系。在随机分组后，1464张网页图像没有被分入任何一个小组内。这些图像被留作验证集合，它们的作用将在个性化推荐系统部分（第8.3.5.6 节）叙述。

为了用户调研的方便，我们进一步将每个小组分为$U = 10$个子组（subgroups），分别邀请U个志愿者进行标注。随后我们在每个子组内随机挑选一半的网页对用于训练并且将剩余的用作测试，建立起独立的训练和测试网页分割集合。这样，我们保证了每个训练集和测试集中都包含了各个受试者均等数量的标注网页对，也就是说，每个受试者的喜好可以在每次测试中都得到平等的加权对待。

记$\mathcal{S}_{t,u}$为$\mathcal{G}_t$的第u 个子组，由第u个受试者标注。所有由第u个受试者标注的子组，就构成了他/她的标注集：$\mathcal{A}_u = \bigcup_t \mathcal{S}_{t,u}$，$u = 1, 2, \ldots, U$。$\mathcal{A}_u$中包含的网页对数量为：$M' = (M/U) \times T = 2000$。

记$\mathcal{S}_{t,u}$中的训练和测试集合分别为$\mathcal{S}_{t,u}^{\mathrm{TR}}$和$\mathcal{S}_{t,u}^{\mathrm{TE}}$，这样就有：$\mathcal{S}_{t,u} = \mathcal{S}_{t,u}^{\mathrm{TR}} \cup \mathcal{S}_{t,u}^{\mathrm{TE}}$。$\mathcal{G}_t$的训练和测试集合分别定义为：$\mathcal{G}_t^{\mathrm{TR}} = \bigcup_u \mathcal{S}_{t,u}^{\mathrm{TR}}$，$\mathcal{G}_t^{\mathrm{TE}} = \bigcup_u \mathcal{S}_{t,u}^{\mathrm{TE}}$。我们还可以类似地构建$\mathcal{A}_u$的训练和测试集合$\mathcal{A}_u^{\mathrm{TR}}$和$\mathcal{A}_u^{\mathrm{TE}}$：$\mathcal{A}_u^{\mathrm{TR}} = \bigcup_t \mathcal{S}_{t,u}^{\mathrm{TR}}$，$\mathcal{A}_u^{\mathrm{TE}} = \bigcup_t \mathcal{S}_{t,u}^{\mathrm{TE}}$。

8.3.3.3　用户调研

我们邀请了$U = 10$名志愿者，并且让第u名志愿者标注集合$\mathcal{A}_u$，$u = 1, 2, \ldots, U$。对于每个网页对，我们要求受试者做出判断：根据他/她的喜好，哪一张网页更加美观？为了让**美观**的含义更加清楚，我们引导受试者选择其中他

更愿意张贴在家中起居室的图片。由于$\mathcal{A}_u$由不同$\mathcal{G}_t$的大小相同的子组构成，每个受试者都能够看到具有不同排名差的网页对。 $\mathcal{A}_u$内的网页对顺序经过随机打乱，受试者每次在屏幕上只能看到一个网页对。每个受试者的测试时间大约为2小时。在标注过程中，我们不告诉受试者网页的Alexa排名信息。

由于对美观性的判断是非常主观的，很多时候受试者本身也无法确定自己的决定。为了最大程度减少噪声，我们要求每个受试者在同一个集合$\mathcal{A}_u$上标注三轮。每一轮内，网页的顺序都经过不同的随机打乱，而且相邻两轮的间隔在两天以上。在第一轮过后，受试者被要求写下一小段评论，解释他们标注的原则。在第二轮和第三轮标注之前，我们请受试者回顾他们写下的话，并且尽量复现他们第一轮的标注结果。如果受试者在某个网页对上的标注在三轮内完全相同，我们认为此标注具有**高置信度**，否则我们选择出现两次的标注结果，并且认为它具有**低置信度**。 **我们根据高置信度标注的比例奖励受试者，以激励他们产生高质量的标注集合。** 对于高置信度和低置信度的情况，我们都接受标注，然而对于后者分配较低的权值以降低噪声的影响。所有受试者标注的所有网页对中，高置信度比例为60%（随机猜测将得到25%的比例）。

经过用户标注后，网页对$\mathbf{p}_{t,m}$就被扩充为：$\mathbf{p}_{t,m} = (a_{t,m}, b_{t,m}, z_{t,m}, u_{t,m}, c_{t,m})$。其中，$u_{t,m} \in 1, 2, \ldots, U$是受试者的ID，而$z_{t,m} \in \{+1, -1\}$表示受试者在这一网页对上的决定：如果他/她认为$\mathbf{I}_{a_{t,m}}$比$\mathbf{I}_{b_{t,m}}$更美观，则$z_{t,m} = 1$。 $c_{t,m}$是标注的置信值：1表示高置信度标注，0.5表示低置信度标注。

8.3.3.4　流行度和美观度的相关性

对于每个标注的网页对$\mathbf{p}_{t,m}$，我们检查流行度和美观度的判断结果是否吻合，即用户是否更加喜欢在Alexa上排名更高的网页：$z_{t,m} = \mathbb{I}_{a_{t,m} > b_{t,m}}$。图8.9展示了统计结果。能够看出，Alexa排名和用户调研之间的相关性比较弱，表现为较低的统计显著性（p值仅为0.1922）。即使在最后一组$\mathcal{G}_{20}$中（Alexa的排名差值非常大），仍然有较多的（100% – 47.8% = 52.2%）低排名网页受到了受试者的偏好。

产生这种现象的原因可能是两方面的。第一，大部分网页的流行是源于它们提供的服务。例如`www.reddit.com`（第25名）和`www.craigslist.org`（第63名）都是非常流行的网站，但是它们的排版却不甚美观。第二，每名用户都有自己的审美偏好。在图8.8的最后一个例子中，受试者#1相比于`www.360.cn`（第33名）更喜欢`www.cloudprohosting.com`（第18438 名）。根据该受试者提供的说明，他/她并不喜欢前者的颜色和排版，而喜欢后者包含的大幅图片。

ALEXA排名组 \ 受试者	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	AVG
#1	39.0	46.0	53.0	53.0	66.0	55.0	53.0	59.0	48.0	41.0	51.3
#2	55.0	41.0	52.0	51.0	57.0	56.0	48.0	45.0	46.0	46.0	49.7
#3	52.0	52.0	65.0	52.0	49.0	50.0	50.0	55.0	49.0	47.0	52.1
#4	53.0	46.0	48.0	51.0	43.0	37.0	55.0	50.0	58.0	52.0	49.3
#5	58.0	48.0	46.0	45.0	48.0	53.0	47.0	41.0	50.0	54.0	49.0
#6	52.0	48.0	49.0	47.0	54.0	51.0	56.0	49.0	46.0	61.0	51.3
#7	57.0	52.0	50.0	54.0	42.0	53.0	49.0	51.0	48.0	45.0	50.1
#8	52.0	67.0	56.0	46.0	54.0	42.0	59.0	52.0	53.0	50.0	53.1
#9	48.0	48.0	47.0	42.0	47.0	48.0	52.0	52.0	49.0	50.0	48.3
#10	40.0	46.0	42.0	47.0	53.0	56.0	51.0	51.0	51.0	56.0	49.3
#11	54.0	55.0	49.0	60.0	47.0	49.0	49.0	48.0	47.0	56.0	51.4
#12	56.0	52.0	41.0	54.0	47.0	47.0	53.0	38.0	51.0	52.0	49.1
#13	50.0	50.0	50.0	51.0	41.0	43.0	59.0	51.0	54.0	45.0	49.4
#14	47.0	54.0	48.0	56.0	55.0	43.0	54.0	50.0	45.0	51.0	50.3
#15	62.0	55.0	54.0	50.0	44.0	43.0	47.0	49.0	47.0	53.0	50.4
#16	56.0	50.0	48.0	50.0	51.0	59.0	45.0	38.0	40.0	58.0	49.5
#17	53.0	48.0	61.0	51.0	48.0	58.0	53.0	57.0	53.0	48.0	53.0
#18	44.0	53.0	55.0	58.0	53.0	52.0	53.0	51.0	51.0	45.0	51.5
#19	55.0	43.0	45.0	55.0	49.0	48.0	48.0	54.0	53.0	53.0	50.3
#20	47.0	45.0	53.0	51.0	46.0	50.0	46.0	40.0	53.0	47.0	47.8
AVG	51.5	50.0	50.6	51.2	49.7	49.7	51.4	49.1	49.6	50.5	50.3

图 8.9　每一个子组的100对网页中，用户更加喜好的网页也在Alexa上排名更高，即流行度与美观度吻合的比例（%）。

8.3.4　我们的算法

为了探索单纯视觉内容与网页的流行度和美观度之间的联系，我们将问题建模为一个图像二分类的任务，并使用现有算法进行处理，包括基于若干种特征的模型（第8.3.4.1 节），以及一个基于卷积神经网络（CNN）的模型（第8.3.4.2节）。

在用户调研环节，我们可能同时得到具有强置信度和弱置信度的标注（见第8.3.3.3 节）。为了使弱置信度的标注信息被相应地减弱，我们对训练和测试过程都做出相应的调整。在基于特征的模型训练中，我们将弱标注对应的特征向量乘以0.5；在基于CNN的模型训练中，我们将弱监督信号产生的错误信号乘以0.5。在测试过程中，我们将弱置信样本对应计数为0.5 而不是1。如果有N_1个强置信样本，其中M_1个被正确预测，同时有N_2个弱置信样本，其中M_2个被正确预测，那

么修改后的预测准确率为：$(M_1 + 0.5M_2) / (N_1 + 0.5N_2)$。无论利用何种预测模型，预测准确率都先在子组内进行计算，然后逐渐累加到更大的单元中去。对于每个子组，我们随机产生10次训练和测试切分，并且报告平均的预测准确率。

8.3.4.1　基于特征的分类模型

基于特征的模型将每张图像$\mathbf{I}_n$表示为一个长向量$\mathbf{f}_n$，并且使用机器学习工具（例如SVM）进行训练和测试。特征抽取过程将在随后详细讨论。

我们的目标是学习一个二分类函数：$r(\mathbf{f}_a, \mathbf{f}_b) \in \{+1, -1\}$，以预测给定网页对中抽取的图像特征$\mathbf{f}_a$和$\mathbf{f}_b$。为简单起见，我们考虑线性核函数的简单情形：$k_\star = k(\mathbf{w}, \mathbf{f}_\star) = \langle \mathbf{w}, \mathbf{f}_\star \rangle$，其中$\mathbf{w}$是需要学习的参数。对比两个核函数就能够得到预测结果：$r(\mathbf{f}_a, \mathbf{f}_b) = \mathrm{sgn}(k_a - k_b) = \mathrm{sgn}(\langle \mathbf{w}, \mathbf{f}_a \rangle - \langle \mathbf{w}, \mathbf{f}_b \rangle) = \mathrm{sgn}(\langle \mathbf{w}, \mathbf{f}_a - \mathbf{f}_b \rangle)$。因此我们可以将向量$\mathbf{f}_a - \mathbf{f}_b$当成一个独立的样本，并且使用二分类器预测分类结果。

实际操作中，对于每个训练网页对$\mathbf{p}_{t,m} = (a_{t,m}, b_{t,m})$，不失一般性，我们可以假设网页$\mathbf{I}_{a_{t,m}}$比$\mathbf{I}_{b_{t,m}}$的质量高（排名靠前或者受到用户更多的偏好）。我们随即计算特征向量$\mathbf{f}_{a_{t,m}}$和$\mathbf{f}_{b_{t,m}}$，并且分别产生差向量$\mathbf{f}_{a_{t,m}} - \mathbf{f}_{b_{t,m}}$和$\mathbf{f}_{b_{t,m}} - \mathbf{f}_{a_{t,m}}$作为正负训练样本。所有的训练样本都被送入一个线性SVM[99]进行处理。我们产生上述对称的训练样本，以抵消SVM的偏置参数（常数项），这样在测试过程中，我们就无需产生对称的训练样本了。

我们抽取多种有效的视觉特征，即**纹理**、**颜色**、**融合**、**布局**、**场景**和**深度**特征。

- **纹理**特征通过将SIFT描述子在BoVW 模型上量化获得[19]。图像首先被重置为320 × 180像素。我们使用VLFeat库[174] 提取密集的RootSIFT[118]描述子。密集采样的空间跨度和窗口大小分别设置为6和12。 SIFT描述子的维度通过主成分分析（PCA）从128降至64。我们训练一个具有32个分量的高斯混合模型（GMM），并且使用改进的Fisher向量（IFV）[7]进行特征编码。最后，我们采用平均池化（average-pooling）以及一个具有4个区域（整张图像以及三个均分的水平条）的空间金字塔。得到的16384维特征向量将先后进行平方归一化和ℓ_2范数归一化[180]处理。 **纹理**特征是视觉识别系统中最常用的高维特征之一。
- **颜色**特征通过将**纹理**特征中的SIFT描述子替换为LCS描述子获得。局部颜色统计量（Local Color Statistics，LCS）[7]是一种96维颜色描述子，通过计算图像在红、绿、蓝三个信道上的均值和方差，捕捉颜色特征。 **颜色**特征的维度同样是16384。

- **融合**特征通过将**纹理**和**颜色**特征拼接起来而获得。**融合**特征的长度为$16384 \times 2 = 32768$。
- **布局**特征通过在图像的布局图（layout map）上计算类似**颜色**特征而获得。布局图是与原图大小相同的灰度图像，其上每一个像素表示该点在原图上与周围环境的差异程度。像素的差异程度可以通过积累原图上像素灰度值与周围的差别，并使用高斯加权获得。剩余的计算过程与**颜色**特征相似，只不过使用单信道LCS描述子（32维，不使用PCA进行量化）。**布局**特征的维度，在使用具有4个区域（整张图像以及三个均分的水平条）的空间金字塔后，为8192。
- **场景**特征对应于GIST描述子[235]。原始的GIST描述子具有512维。我们采用具有5个区域（整张图像以及2×2网格）的空间金字塔，使得**场景**特征的维度增加至$512 \times 5 = 2560$。
- **深度**特征来源于在大规模视觉识别任务上预训练的深度卷积神经网络[11]的中间输出。这些特征被证明在许多情况下具有很强的可推广性和可迁移性[138][128]。图像大小被重置为227×227，并且作为**Alex-Net**[11]的输入。我们提取倒数第二个全连接层的输出信号（通常称为*fc-6*）。**深度**特征的维度为4096。

对于上述所有特征（除了**深度**特征），我们同时测试使用或者不使用SPM方法的版本。

8.3.4.2 基于CNN的分类模型

我们同时提供一种基于深度卷积神经网络（Convolutional Neural Networks，CNN）的分类方法。

网络的整体结构如图8.10所示。它由两条独立的单图像处理链构成，并且包含一个**比较神经元**（**comparison neuron**）用于输出信号。每条图像处理链的输入都是一个47×47的RGB图像（随后解释）；而整个网络的输出是一个浮点数，表示左边的图像更好（流行度或者美观度更强）的置信程度。在分开的处理链中，每张图像经过四个单元的处理。在每个单元中，图像数据先经过一些3×3卷积核的处理（图像四周被拓宽1个像素以保证其长宽不变），随后在一些大小为3×3、空间跨度为2（除了最后一层5×5全局池化）的窗口中被最大池化（max-pooling）。我们简单地将所有隐藏层（hidden layers）的卷积核数量都固定为128。我们使用矫正的线性单元（ReLU）[11]作为卷积后的激励函数，并且在除了最后一个池化单元外的所有池化单元之后应用dropout技术[129]（30%丢弃率）。分开的链式网络结构在小图像识别任务中被证明非常有效：它们能够在MNIST和CIFAR-10数据集

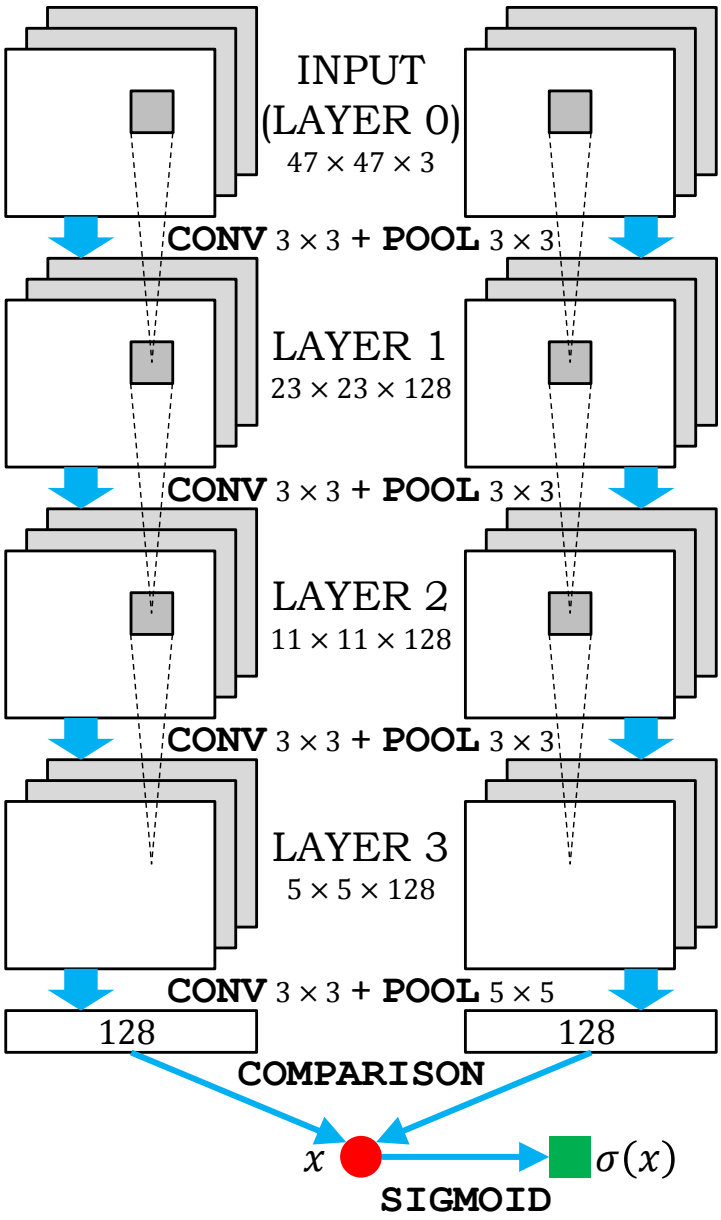


图 8.10　用于网页预测的深度卷积神经网络模型。

上分别得到0.36%和7.15%错误率。因此，我们可以认为两张图像都能够被倒数第二层的128维向量有效地表示出来。

比较神经元是我们的网络中一个重要的特点。它的目标是产生一个浮点数x，表示左边图像比右边图像更好的置信程度（介于$(0,1)$之间）。假设倒数第二层产生的向量表示分别为$\mathbf{w}_a$和$\mathbf{w}_b$，那么对比函数就能够写成：$x = f(\mathbf{w}_a, \mathbf{w}_b)$。实际操作中，我们简单地使用两个线性核函数来计算$x$：$x = f(\mathbf{w}_a, \mathbf{w}_b) = \langle \mathbf{w}_a, \mathbf{x}_a \rangle + \langle \mathbf{w}_b, \mathbf{x}_b \rangle$。其中，$\mathbf{w}_a$和$\mathbf{w}_b$是比较权值，它们的初值都设为$\mathbf{0}$向量，并且在学习过程获得调节。最后，$x$被当成一个S形函数（sigmoid函

数）的输入：$\sigma(x) = \frac{e^x}{1+e^x}$。 $\sigma(x)$就是整个网络的输出。

在训练过程中，正（左边图像更好）负训练样本分别具有监督信号$\sigma = 1$和$\sigma = 0$。 CNN的训练过程即向后传递错误信息并且相应更新网络参数。通过计算$\frac{\partial f}{\partial \mathbf{x}_a}$、$\frac{\partial f}{\partial \mathbf{w}_a}$、 $\frac{\partial f}{\partial \mathbf{x}_b}$和$\frac{\partial f}{\partial \mathbf{w}_b}$，比较神经元的参数学习过程与卷积层非常类似。在测试过程中，预测结果为左边图像更好，当且仅当$\sigma > 0.5$。

我们最后阐述网络训练的细节。对于每个网页对，图像首先被重置为96×54像素（长宽比不变，按1 : 20缩小）。为了进行数据扩张[11]，我们每次从96×54原图上随机切割47×47子图。由于96×54图像包含400种不同的47×47子图，一个网页对可以扩张为$400^2 = 160$K个不同的子图对。我们的训练网页对总量为500（Alexa排名小组$\mathcal{G}_t$）或者1000（用户标注小组$\mathcal{A}_u$），经过数据扩张，不同的子图对数量就分别达到了80M和160M。我们采用三阶段训练过程，每个阶段的训练子图对数分别为10^7、10^6、10^6，而学习率分别为0.001、0.0001、0.00001。在一个NVIDIA GeForce Titan-GPU上，训练过程大约花费0.7小时。测试过程中，我们并不枚举一个网页对所对应的全部子图对，而是简单地测试400对位置相同的子图对。 400个网络响应值经过平均，就得到了最后的测试结果。在Titan-GPU上，测试一对网页（400个窗口对）需要大约0.08秒。

8.3.5　实验部分

本小节研究基于视觉的网页流行度/美观度预测问题，并且提出两个新的商业应用：**网站流行度趋势预测**和**个性化网页推荐**。

8.3.5.1　流行度预测

我们利用视觉信息，预测由Alexa排名定义的网页流行度。我们首先将测试限制在一个小组$\mathcal{G}_t$ 内，其中网页对的排名差限制在$(1000t, 1000\,(t+1)]$范围内。在每个小组中，500对网页被用于训练，其余被用于测试。

预测结果如图8.11 所示。可以观察到，在靠前的几个小组中（网页排名差较小，例如$\mathcal{G}_1$、$\mathcal{G}_2$和$\mathcal{G}_3$），网页流行度的预测准确率接近50%（随机猜测水平）。随着排名差的增加，我们观测到了更高的预测准确率。在最后一个小组（$\mathcal{G}_{20}$）内，甚至不止一种模型得到了超过90%的准确率。这说明视觉信息确实与Alexa排名具有相关性，通过视觉特征将高排名和低排名的网页区分开来是有可能的。

为了揭示靠前的若干小组中预测准确率较低的原因，我们观察2014年8月25日以及2014年9月23日的Alexa排名数据。这也是我们连续30天数据收集过程的头尾两天。我们注意到，一些网页的Alexa排名随着时间的变化非常剧烈。

		ALEXA排名组																				
		#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16	#17	#18	#19	#20	AVG
图像特征	纹理	50.7	50.6	51.0	51.7	52.5	51.7	52.8	50.1	52.0	54.5	57.3	57.0	57.2	58.8	59.0	60.0	65.2	66.7	71.9	86.9	57.9
	纹理+S	50.9	50.0	49.6	53.0	51.4	51.8	51.9	51.5	53.6	53.3	57.2	58.3	58.6	60.0	61.4	62.9	66.9	69.4	75.2	91.4	59.0
	颜色	50.8	50.4	48.8	53.6	51.7	52.7	**53.2**	51.7	53.1	53.6	54.8	55.9	57.2	58.3	60.2	59.4	62.5	67.6	72.3	84.9	57.6
	颜色+S	51.0	49.9	50.3	51.9	52.2	52.0	51.9	51.7	52.9	52.9	56.7	57.6	58.9	58.5	61.3	62.6	64.6	70.1	74.4	89.0	58.5
	融合	50.8	50.7	48.7	53.2	52.6	52.3	52.5	51.9	53.4	54.1	57.1	58.6	59.0	60.5	63.4	61.7	65.7	70.1	75.4	88.7	59.1
	融合+S	51.0	49.2	49.1	53.2	51.7	52.4	52.6	51.9	53.9	54.0	56.9	59.1	**61.2**	**61.1**	**64.7**	**64.9**	**68.6**	**71.8**	**77.6**	**91.5**	**59.8**
	场景	50.2	51.3	50.1	48.4	**53.4**	51.5	50.3	51.2	52.4	49.6	52.3	52.6	54.0	52.6	55.7	54.1	57.7	58.2	59.8	71.2	53.8
	场景+S	51.2	**51.7**	**51.3**	48.2	52.9	50.0	51.9	51.3	52.5	49.8	52.6	52.5	53.4	52.1	55.2	55.7	56.0	57.3	61.6	69.3	53.8
	布局	**51.6**	50.2	50.7	**57.2**	51.8	52.6	51.1	50.4	54.3	50.5	53.0	55.0	52.7	52.0	55.7	56.2	58.8	61.0	67.4	81.8	55.7
	布局+S	49.9	50.0	49.6	55.0	51.1	**53.6**	52.2	51.7	**54.9**	51.6	53.7	56.3	56.0	56.7	60.4	60.7	59.5	63.3	72.1	86.8	57.3
	深度	50.8	49.4	50.3	53.3	52.2	51.9	51.7	51.7	54.7	**54.8**	**57.2**	59.2	60.6	58.6	62.5	63.7	64.2	70.0	77.0	91.1	59.3
	CNN	50.6	49.7	50.5	52.9	51.8	52.4	53.1	**52.7**	53.7	53.9	56.3	**59.8**	60.1	60.6	62.8	62.6	63.9	68.6	76.9	89.8	59.1
	AVG	50.8	50.3	50.0	52.6	52.1	52.1	52.1	51.5	53.5	52.7	55.4	56.8	57.4	57.5	60.2	60.3	62.7	66.2	71.8	85.2	57.6

图 8.11　不同的分类模型在不同的Alexa排名小组中产生的预测准确率（%）。**加粗且下划线**的数字表明该小组的最佳预测准确率。可以看出，预测准确率随着网页排名差的增加而上升。

以www.bloglovin.com为例，该网站在第一天的排名为第500名，然而在最后一天却跌落至第846名，相对变化接近70%；另一个例子来源于www.wowwiki.com，该网站第一天排在第10000名，却在30天内被提升至第5726名，相对变化接近43%。另一方面，网页的视觉内容，尤其是网页的风格和排版，却不会在短时间内发生较大的变化。尽管我们使用的排名数据已经在30天范围内取平均值，仍然普遍存在的不稳定性使得我们难以对Alexa排名相近的网页对做出准确预测的原因之一。

一个额外的线索来自Alexa排名小组的交叉预测实验。为此，我们在一个小组的训练集合$\mathcal{G}_i^{\mathrm{TR}}$上训练分类器，并且在另一个小组的测试集合$\mathcal{G}_j^{\mathrm{TE}}$上进行评估，得到预测准确率$p_{i,j}$。由此得到的准确率矩阵$\mathbf{P} = \left[p_{i,j}\right]_{T\times T}$如图8.12 所示。我们可以看到，每一个测试集上的最高预测准确率通常都来源于较大排名差小组上的训练，即高亮的数字通常出现在对角线上或者其下部。例如，在$\mathcal{G}_{19}^{\mathrm{TR}}$上训练而在$\mathcal{G}_7^{\mathrm{TE}}$上测试，得到$p_{19,7} = 57.6\%$，比$p_{7,7} = 51.9\%$要高许多。这种现象表明，存在一种相对稳定的视觉特征，能够大致预测网页的Alexa排名；同时，在包含噪声更少的小组中通常能够训练更加稳定的分类模型。如果训练数据过于不稳定，模型即便在最简单的测试集上，也无法得到满意的效果，例如$p_{1,20} = 51.5\%$而$p_{20,20} = 91.4\%$。

8.3.5.2　美观度预测

接着，我们预测通过用户调研标注的网页美观度。我们继承了前面实验的设定，只是简单地将Alexa排名小组$\{\mathcal{G}_t\}$替换为受试者标注小组$\{\mathcal{A}_u\}$。图8.13展示

训练ALEXA排名组 \ 测试ALEXA排名组	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15	#16	#17	#18	#19	#20	AVG
#1	50.9	51.1	50.2	52.4	50.3	49.2	48.9	50.8	51.0	50.1	50.8	49.2	52.1	49.3	50.8	51.0	51.6	51.6	52.4	51.5	50.8
#2	48.1	50.0	52.4	50.6	49.3	50.8	50.0	51.3	48.6	49.2	52.5	51.9	47.9	52.3	52.1	52.3	51.5	54.3	55.1	55.2	51.3
#3	49.7	50.3	49.6	50.6	50.4	49.4	49.9	49.3	50.5	50.9	52.1	51.4	49.9	51.8	53.0	53.0	52.5	52.1	54.2	54.6	51.3
#4	51.0	48.4	50.7	53.0	51.7	51.3	50.1	51.6	50.6	51.3	51.4	49.6	53.3	51.2	52.8	52.8	53.7	56.6	56.0	54.4	52.1
#5	**51.7**	50.2	49.3	49.9	51.4	51.8	51.2	51.9	51.2	51.6	52.3	52.6	53.2	51.6	53.1	57.3	54.0	54.7	53.9	56.6	52.5
#6	50.5	51.4	48.2	51.2	54.0	51.8	51.9	51.2	50.7	51.4	51.9	53.1	53.1	53.7	53.4	54.6	54.2	54.3	55.0	56.8	52.6
#7	48.6	51.2	51.3	51.8	51.9	51.2	51.9	54.6	53.2	51.9	55.5	54.9	56.3	55.7	56.4	57.3	56.5	56.8	59.2	59.1	54.3
#8	51.5	50.6	49.9	50.8	51.5	50.1	53.2	51.5	53.0	51.1	54.6	54.3	54.0	55.0	55.6	54.1	55.9	54.9	54.6	59.3	53.3
#9	51.2	48.6	51.6	53.0	52.0	52.2	52.5	51.8	53.6	52.9	54.6	54.3	57.9	54.9	55.5	55.4	55.1	57.2	58.4	58.1	54.0
#10	49.8	48.6	49.5	53.5	51.6	50.9	53.1	51.9	52.9	53.3	54.8	57.3	55.9	57.3	54.3	57.2	57.0	55.2	52.9	54.6	53.6
#11	51.4	50.2	50.7	51.0	52.9	51.0	54.9	56.4	54.5	54.5	57.2	56.8	58.7	58.8	57.4	61.0	58.8	58.1	59.7	61.1	55.8
#12	49.5	50.8	51.8	52.3	52.5	52.6	53.5	54.0	54.4	**57.3**	57.8	58.3	57.1	57.5	59.0	58.6	58.1	58.9	58.8	61.2	55.7
#13	50.8	49.6	49.0	51.5	51.8	53.8	53.9	53.1	56.2	55.5	57.9	58.3	59.6	57.6	59.3	61.6	60.9	59.2	61.3	60.8	56.1
#14	49.7	51.6	51.6	50.6	51.8	53.9	55.2	55.5	55.5	55.2	58.6	57.8	60.2	60.0	60.8	60.1	61.3	60.2	59.6	60.9	56.5
#15	48.8	50.8	52.0	53.2	54.4	54.9	54.1	53.9	55.1	55.1	58.9	58.5	59.3	58.6	61.4	61.0	61.5	62.2	63.4	66.7	57.2
#16	51.2	49.8	52.3	53.3	**56.0**	**55.7**	55.7	54.2	54.9	56.5	**59.0**	**58.7**	**61.3**	59.5	61.1	62.9	63.8	64.5	65.3	66.3	58.1
#17	49.2	50.3	**52.7**	54.5	55.2	55.4	56.0	**56.9**	55.2	56.2	58.0	58.4	59.3	**60.4**	60.4	64.5	**66.9**	64.9	67.5	68.6	58.5
#18	49.6	51.5	51.9	55.0	54.9	54.7	55.3	54.1	55.4	54.7	57.1	57.9	59.0	60.1	62.5	**66.5**	65.4	69.4	71.2	74.1	59.0
#19	50.7	**53.1**	51.8	**55.5**	53.2	55.0	**57.6**	55.9	**58.8**	53.7	57.6	57.6	59.6	58.5	62.1	62.9	65.6	**70.7**	75.2	82.6	59.9
#20	50.9	52.7	51.5	53.9	54.7	54.0	54.8	56.5	56.1	54.7	57.2	56.4	59.4	58.6	**63.1**	62.1	64.1	70.0	**80.9**	**91.4**	**60.2**
AVG	50.2	50.5	50.9	52.4	52.6	52.5	53.2	53.3	53.6	53.4	55.5	55.4	56.4	56.1	57.2	58.3	58.4	59.3	60.7	62.7	55.1

图 8.12　所有20个Alexa排名小组上的交叉预测准确率（%），由**纹理**特征配合SPM产生。**加粗且下划线**的数字表明该测试小组的最佳预测准确率。

了预测结果。可以观察到，所有特征的预测准确率（对10名受试者取平均）都高于50%（随机猜测水平）。即使对于预测精度最低的特征，即产生50.8%平均准确率的**场景**特征（配合SPM），空假设（10000次随机猜测）的置信p值也显著地小于0.01。这说明视觉特征和网页美观度之间确实存在某种相关性，只不过当前特征不能很好地捕捉这种相关性。

为了解释实验得到的较低的预测准确率，我们首先注意到美观性判断具有很强的主观性[244]。根据神经科学提供的线索[246]，美学判断在人类大脑中的处理机制非常复杂。人们同样相信，美学感知与内脏知觉，特别是对负面效应（反感、疼痛等）的知觉紧密相关[247]。然而，我们设计的预测模型只考虑了低层视觉特征。虽然这些模型能够有效地对客观而具体的视觉概念（如场景或物体）进行识别，但它们通常无法捕捉高级的“情感”特征以进行美学判断。为了克服这一缺陷，我们在第8.3.5.6 节中提出一种个性化网页推荐系统。

图像特征	测试受试者组										
	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	**AVG**
纹理	51.2	52.4	52.4	51.5	51.3	51.8	51.1	50.7	50.8	51.0	51.4
纹理+S	52.3	52.0	52.9	50.6	52.0	52.9	51.4	52.3	51.3	52.9	52.1
颜色	51.5	52.1	51.7	50.8	50.4	50.7	52.1	51.2	50.8	51.7	51.3
颜色+S	51.2	52.4	51.3	50.4	51.9	52.5	**53.3**	51.7	50.3	52.1	51.7
融合	51.8	52.3	52.4	51.3	50.8	52.0	51.8	52.3	50.8	52.2	51.8
融合+S	52.3	52.6	52.5	50.3	51.5	52.4	52.2	**53.2**	50.3	52.8	52.0
场景	51.1	51.1	51.5	51.7	50.3	52.8	50.0	50.1	49.8	50.4	50.9
场景+S	50.0	51.0	51.4	51.5	50.0	52.7	50.8	49.7	49.4	50.9	50.8
布局	49.9	52.6	50.9	**52.2**	51.3	51.7	50.1	49.5	49.8	51.1	50.9
布局+S	53.5	51.8	51.9	51.9	52.0	53.0	52.0	50.6	50.7	51.2	51.8
深度	**53.6**	51.5	53.0	50.8	51.9	**53.6**	52.5	51.8	52.0	**53.5**	**52.4**
CNN	52.8	**52.8**	**53.3**	51.7	**52.4**	52.1	52.7	50.3	**52.5**	53.1	52.0
AVG	51.8	52.0	52.1	51.2	51.3	52.4	51.7	51.1	50.7	51.9	51.6

图 8.13　不同的分类模型在不同的受试者的标注数据中产生的预测准确率（%）。**加粗且下划线**的数字表明该小组的最佳预测准确率。虽然对于每种特征，在10个受试者上的平均准确率相对较低，然而对于每名受试者，都能找到至少一种产生较好预测准确率（不低于52%）的特征。

8.3.5.3　讨论

我们首先观察预测结果与不同模型和特征的关系（见图8.11和图8.13）。在流行度预测和美观度预测任务中，**纹理**和**深度**特征以及**CNN**模型都得到了非常好的总体准确率，而**场景**特征得出的预测结果则较为无法令人满意。此外，我们从使用**深度**特征和直接训练**CNN**模型得到的结果非常类似，说明两种基于深度学习的方法可能捕捉到了非常相似的视觉线索。

在第8.3.3.4 节中，我们发现网页的流行度和美观度之间不具有强相关性。为了更好地解释这一现象，我们使用10个受试者的标注集$\mathcal{A}_u^{\mathrm{TR}}$训练了10 个分类器（使用**深度**特征），并且将它们用于预测Alexa排名小组$\mathcal{G}_{20}^{\mathrm{TE}}$。我们知道，$\mathcal{G}_{20}$是一个非常容易预测的小组：在$\mathcal{G}_{20}^{\mathrm{TR}}$上训练的**深度**特征模型在$\mathcal{G}_{20}^{\mathrm{TE}}$ 上产生了91.1%的准确率。然而，10个分类器中甚至没有一个能够达到55%以上的预测精度，最差的准确率甚至只有49.4%（在$\mathcal{A}_9^{\mathrm{TR}}$上进行训练，比随机猜测更差）。这一实验再次表明，预测流行度和美观度，需要训练不同的模型参数。这也验证了人们在日常生活中的行为：大多数人访问网站的目的并非因为它美观，而是因为它切合了自身的需求。

8.3.5.4　用户喜好研究

为了发现为什么不同受试者的喜好需要通过不同的特征进行预测，我们参考受试者们在第一轮标注过后写下的小结，尝试寻找他们的喜好和特征之间的关系，并且将其中一些有趣的结果列在下面：

- **受试者#1**和**受试者#6**更喜欢那些包含大幅图片（如自然景观）或者呈现美味食物图片的网页，却不喜欢含有大量广告，尤其是广告语言与网页的主体语言不一致的网页。自然地，**深度**和**布局**特征能够很好地预测他们的喜好，因为**深度**特征捕捉了景观和食物等视觉线索，而**布局**特征能够有效地发现网页中的大幅图片。
- **受试者#2** 不喜欢具有唐突广告的网页。他喜欢那些经过精心排版，并且装饰有美丽标志、按钮、气泡以及鲜艳背景颜色的网页。由于广告和各种网页组件都能够被**纹理**特征描述，而**颜色**特征对于背景颜色非常敏感，因此结合了两者的**融合**特征就能够很好地预测**受试者#2**的喜好。
- **受试者#4**在整个测试过程中将眼镜摘去，他希望通过更加结构化的性质而不是细节特征来判断网页的优劣。他不喜欢那些高度不对称的网页，即大部分组件被放置在网页的一边，而另一边相对较空。因此**布局**特征在这种情况下预测得比较好。
- **受试者#9**声明，在绝大部分情况下，他都无法有把握地进行预测。统计上看，他标注的高置信度比例是所有受试者中最低的（55%，平均值约为60%）。当他不知道如何选择的时候，他一般更倾向于选择呈现在屏幕左边的网页。这种情况下，我们几乎无法找到任何一种特征，能够对他的行为进行准确的预测。 **融合**特征和**深度**特征以及**CNN**模型能够比其他方法工作得更好，仅仅因为它们捕捉了更多的视觉信息。

为了观察用户的喜好如何影响模型的训练过程，我们再次进行交叉预测实验。记$q_{i,j}$为在集合$\mathcal{A}_i^{\mathrm{TR}}$上训练而在集合$\mathcal{A}_j^{\mathrm{TE}}$上预测的准确率。准确率矩阵$\mathbf{Q} = \left[q_{i,j}\right]_{U\times U}$如图8.14 所示。一方面，我们同样能够观测到统计显著性：除了受试者#9（高置信度比例最低者），其他受试者都提供了较好的训练数据，从而在10个受试者上的平均测试准确率超过了51%（p值小于0.001）。因此我们得到结论，尽管每个受试者的偏好不同，我们仍然发现一些可学习的视觉知识，能够在不同用户之间迁移。另一方面，同一受试者标注的训练和测试数据一般具有更高的可解释性，即在同一受试者的训练集上进行训练和测试，一般能够得到更高的准确率（矩阵**Q**的对角线元素一般大于相应行列的均值和中位数）。作为比较，矩阵的对角线元素和非对角线元素的平均值分别为52.1%和51.0%。

训练受试者组 \ 测试受试者组	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	AVG
#1	52.3	51.2	51.2	50.1	50.2	51.8	51.2	51.8	**52.2**	53.0	51.5
#2	51.2	52.0	50.6	50.4	52.3	51.5	51.5	50.7	50.7	51.2	51.2
#3	51.0	51.6	**52.9**	51.5	52.5	50.6	52.0	51.1	48.9	**53.3**	51.6
#4	50.1	50.7	52.4	50.6	50.4	52.2	50.9	52.3	51.3	50.6	51.1
#5	51.6	52.1	52.3	50.6	52.0	50.6	51.3	51.2	48.6	51.5	51.2
#6	51.9	51.9	51.6	50.3	**52.8**	**52.9**	51.3	50.7	50.4	51.0	51.5
#7	51.7	51.9	51.1	**51.6**	51.1	51.8	51.4	51.7	50.8	52.7	51.6
#8	52.0	51.7	51.1	51.1	51.3	50.6	50.4	**52.3**	50.8	51.4	51.3
#9	50.9	**52.2**	49.2	50.8	49.4	50.6	49.7	51.8	51.3	50.1	50.6
#10	**53.2**	50.8	52.7	51.1	52.2	52.1	**51.6**	50.8	49.9	52.9	**51.7**
AVG	51.6	51.6	51.5	50.8	51.4	51.1	51.1	51.4	50.5	51.8	51.3

图 8.14　10个用户调研小组上的交叉预测准确率（%）。我们使用**纹理**特征配合SPM方法。**加粗且下划线**的数字表明该小组的最佳预测准确率。对角线上的平均值为52.1%，明显高于整张表的平均值51.3%。

8.3.5.5　网站流行度趋势预测

预测网站的流行度趋势具有很大的商业价值，据此人们就能够决定是否赞助这一网站或者在其上发布广告。我们回到30天内收集的$N = 14910$个网站的Alexa排名数据（见第8.3.3.1 节）。将一个网站r_n每天的Alexa排名记为$(r_{n,1}, r_{n,2}, \ldots, r_{n,30})$，$n = 1, 2, \ldots, N$。我们设定9个检查点，$(d_1, d_2, \ldots, d_9) = (1, 2, 3, 5, 8, 12, 17, 23, 30)$，并且预测在每个$l = 2, 3, \ldots, 9$处，$r_{n,d_l} < r_{n,d_{l-1}}$是否成立，即该网站的排名是否得到了提升。我们设定不均匀的d_l值，以观察预测结果与**预测间隔**（prediction gap，即$d_l - d_{l-1}$）的关系。一半（$N/2$）的网站被随机挑选出来用于训练，而剩余的则用于测试。对于所有的测试日期，训练和测试集的划分是固定的。

我们简单地采用基于特征的分类模型（当然，基于CNN的分类模型也能用于这一任务）。在第d_l天，我们根据$r_{n,d_l} < r_{n,d_{l-1}}$是否成立，将训练网站分为两类，它们对应的图像特征则分别作为正负训练样本。由于Alexa排名每天都会发生变化，我们必须在每次预测时单独训练分类器。

以预测准确率表示的实验结果如图8.15所示。可以看到，采用合适的特征时，3天之内的预测准确率一般高于60%（随机猜测将得到50%）。这说明在相当程度上，视觉特征能够解释网站的短期流行趋势。正如之前实验观察到的一样，**融合**与**深度**特征得到了最好的预测结果。此外，预测精度随着预测间隔增大而下降，

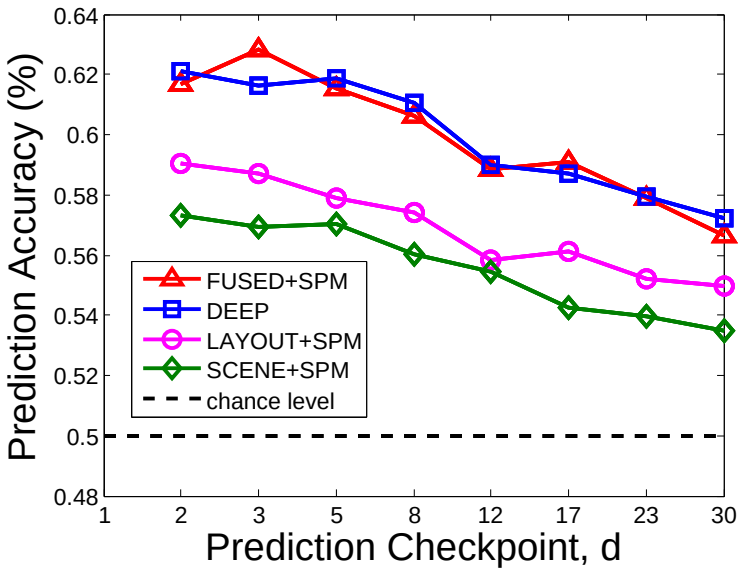


图 8.15　网站流行度趋势预测的准确率与预测间隔（prediction gap）的关系。

说明视觉信息对于网站排名的影响随着时间的推移而下降；由于网站流行趋势受到更多因素的影响，通过视觉特征预测一段较长时间后网站的流行度趋势将更加困难。

8.3.5.6　个性化网页推荐

在第8.3.5.4 节中，我们论述了不同用户可能具有不同的喜好。如果事先了解他们的喜好，就能够极大地辅助我们设计出更加精准捕捉用户需求的系统[244]。这一节里，我们提供一种简单的方法，从用户标注的数据中学习他们的美观度喜好，并且使用训练后的模型为他们提供更针对性的网页推荐。

为此，我们使用在小组$\mathcal{A}_u^{\mathrm{TR}}$上训练的十个用户分类模型，$u = 1, 2, \ldots, U$，用于测试无人标注过的验证集合（1464张网页图像，见第8.3.3.2 节）。我们对比验证集合中的每一对网页（总共$\binom{1464}{2} \approx 10^6$对），得到每张图像的1463个预测分数（+1或者−1，分别表示更好和更差）。我们计算每一张网页得到+1的次数，并依此得到一个排序列表。如果一张网页出现在排序列表的0%–20%、40%–60%和80%–100%范围内，则分别被评分为**好**、**中**和**差**（用户相关）。一些被评估为**好**、**中**、**差**的网页实例如图8.16 所示。

用户调研方面，我们随机选取了每个受试者对应的分类模型产生的200个**好**和200个**差**网页，构成200个**好**-**差**网页对，并且让该受试者同样从每一对中找出自己更喜欢的网页。最后，我们统计用户的评价与自动推荐结果的吻合程度（受试者更喜欢来自**好**组的网页的比例）。利用**深度**特征训练的分类器，10个受试者的

图 8.16　一些个性化的推荐结果。左、中、右：预测为**差**、**中**、**好**的网页实例。注意，`www.digikey.com`同时出现在受试者#1的**差**组和受试者#4的**好**组中，而`www.ask.fm`两次出现在**差**组中。所有展示的**好**组和**坏**组样本都得到了受试者本人的肯定（确实符合他们的喜好）。

平均预测准确率为56.7%，其中最高者达到了59.2%（受试者#6）。这一结果显著地好于第8.3.5.2 节和第8.3.5.4 节中得到的结果。然而，采取相同方法构建的**好**-**中**和**中**-**差**网页对的预测准确率迅速下降至51.5%和51.9%。因此，我们猜测排名中游的样例可能是导致美观度预测准确率较低的主要原因之一。而本小节所述也不失为一种有效的方法，即使在用户标注数据不甚稳定的情况下，也能够为其提供更好的个性化推荐结果。

8.3.6　结论

本节的研究目标是发现网页的视觉内容和它的流行度/美观度的关系。我们将这个问题建模为二分图像分类，收集了一个网页截屏数据集，并且利用Alexa流量排名和用户调研对大量网页对进行了标注。我们使用现有的视觉识别技术来解决这一问题，包括基于特征的图像分类模型和深度卷积神经网络。实验结果揭示了网页视觉内容和它的流行度/美观度之间的关系确实存在，而且视觉特征能够帮助判断网页的质量。我们的算法还为两个商业应用奠定了基础，即预测网页流行度趋势，以及设计个性化的网页推荐系统。

在未来的工作中，我们将会把数据集发布出来，并且逐渐加入更多的图像和标注信息。我们期待越来越多的研究者关注这一开放、有趣而且具有挑战性的问题。

8.4　本章小结

新问题的提出对于科学研究具有重要的意义。本章基于全文研究的内容，提出两个具有挑战性的新问题，不仅是对前文的总结与应用，也是对未来工作的展望。我们希望这两个新问题能够在未来吸引更多的研究兴趣，从而促进计算机视觉领域的应用与发展。

第 9 章　总结与展望

9.1　本文的总结

图像表示是计算机视觉的根本与核心问题之一。在本文中，我们针对基于局部特征的图像表示模型进行了深入的研究，并且提出了若干重要的科学问题。

我们将基于局部特征的图像表示模型拆分为多个模块，包括特征提取、特征编码、特征组合以及在线查询等。进而，我们深入探索每一个模块中现有方法存在的缺陷，并且根据具体问题的性质，提出创新性的解决方案。作为全文的总结和升华，我们开创性地将图像分类和检索模型统一起来，并且提出了两个富有挑战性的新问题。

本文各部分的主要研究内容和创新点总结如下：

- **第3章：具有翻转不变性的局部特征。** 我们从细粒度图像分类的实际问题出发，论述局部特征的翻转不变特性在图像匹配、分类和检索问题中的重要性。随后，我们观察SIFT特征在翻转后的变化，并提出一种对称算子，抵消翻转带来的变化。我们进一步发现，利用SIFT特征的朝向估计，还能够设计出RIDE算法，将翻转不变性推广到其他局部特征上去。 RIDE算法的简洁和有效性使得它能够应用于一系列图像分类和检索问题。
- **第4章：基于几何短语的强化编码算法。** 我们深入探讨了传统的视觉词袋模型存在的若干缺陷，包括局部特征的描述力不足，以及缺乏中层表示结构。针对这些问题，我们提出了三个模块，即抽取互补的局部特征、几何短语池化以及基于边缘的空间加权算法。通过将这三个模块有机地结合起来，我们增强了视觉词袋模型的表示能力，并且得到了更好的分类效果。
- **第5章：针对不同分类问题的特征组合方法。** 我们从空间金字塔匹配（SPM）的一个简单推广入手，观察特征组合（池化）方法对于分类问题的重要性。随后我们针对两个特殊的分类问题，即细粒度物体分类和场景分类，分别设计出层次化部件匹配和朝向金字塔匹配算法，利用问题的特殊性质，抽取具有更强语义信息的池化箱，从而编码更加鲁棒的图像表示向量。其中，层次化部件匹配是近年来基于部件检测的细粒度分类算法的代表，而朝向金字塔匹配则可以推广到更一般形式的分类问题中去。
- **第6章：图像检索的后处理过程。** 我们着眼于大规模图像搜索问题，探寻视觉词袋模型配合倒排表结构，在检索性能上的不足之处。我们观察到：图像

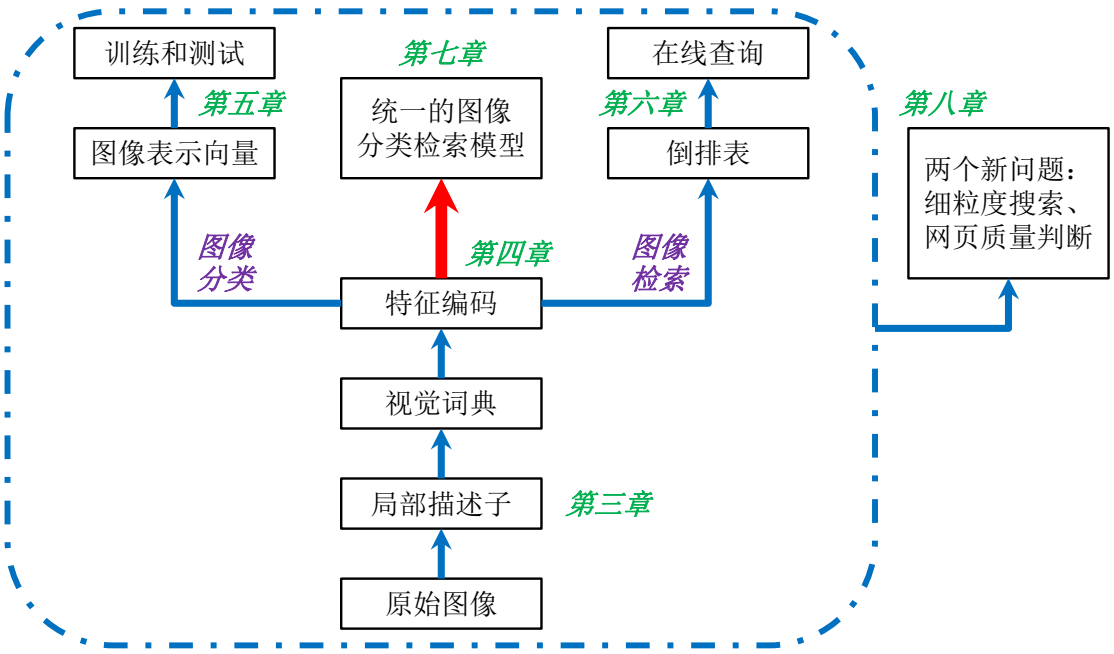


图 9.1　本文的各个部分之间的关系。

与特征之间的关系以及图像与图像之间的关系，蕴含着更加稳定的语义信息，能够辅助检索的后处理过程。因此，我们设计出两种基于置信度传递的算法，即异质图传播算法和图像网络算法，分别利用上述两种关系进行搜索的重排序。其中，后者可以看作前者的一个变种：它的假设更一般，应用范围更广，泛化性能也更好。

- **第7章：统一的图像分类和检索模型。** 传统的图像分类和检索模型通常具有截然不同的在线查询环节。我们开创性地设计出一种算法，通过强大的全局特征和鲁棒的相似度计算，统一了分类和检索模型，并且在两个问题上都取得了优异的性能。为了应对巨大的计算量，我们充分利用GPU的强大计算能力，并且预言这将是未来发展的方向。
- **第8章：两个新问题的探索。** 将前面几个章节的内容结合起来，我们就得到了更加强大的图像表示方法。在这些方法的基础上，我们提出了两个有趣而富有挑战性的问题，即细粒度图像搜索和基于视觉内容的网页质量分析。我们指出这两个问题所蕴含的研究潜力和商业价值，并且希望它们在未来能够得到更多的关注。

我们利用图9.1表示本文各个部分之间的关系。第3章、第4章、第5章和第6章作为四个独立的部分，分别探索了基于局部特征的图像表示模型在各个应用问题上的不足之处，包括图像分类（一般图像分类、细粒度图像分类、场景分类等）以及图像检索（近似重复图像检索和大规模网络图像搜索）。在此基础上，我们

提出了若干针对性算法以解决这些问题。第7章作为上述几个章节的总结和提高，提出了一种统一的算法，用以解决图像分类和检索问题：从全新的角度出发，我们提取更强大的图像描述特征，并且将图像分类和检索模型统一地建模为图像相似度计算的问题。最后的第8章总结了之前的方法，并且提出了两个新问题，以验证视觉算法在实际应用问题中的有效性。

9.2　未来的展望

本文系统地研究了基于局部特征的图像表示模型。我们在解决一些现有问题的同时，也留下了一些问题，以待未来进一步的探索。未来的研究方向，主要集中在视觉词袋模型和深度卷积神经网络的理论分析、对比和结合上。从微观层面看，我们将继续本文中未完成的研究课题，包括细粒度图像搜索的推广和扩展、网页上图像特征的挖掘，等等。从宏观层面看，下面几个根本问题仍将是计算机视觉领域未来的研究重点。

1. **视觉词袋模型的理论分析和表达能力极限。** 视觉词袋模型是一种基于统计的图像表示方法。虽然研究者们对于这一模型的诸多性质已经比较熟悉，然而一些深入的理论分析依然有待展开。例如，如何设计出一种能够将视觉词袋模型的中间表示（如视觉码本）可视化的算法，仍然是一个开放性的问题。此外，视觉词袋模型的表达能力是否有极限，这种极限是否对应着更加强大模型的出现，都有待未来探究。

2. **深度卷积神经网络的理论分析和表达能力极限。** 这一问题与前一问题有很强的对应性。近年来，深度卷积神经网络在许多领域已经超越了视觉词袋模型，逐渐在各类视觉问题中占据了领先地位。然而，卷积神经网络的中间层结果更加难以可视化。特别是在网络不断加深、隐层神经元个数不断增加的情况下，研究者们已经倾向于将深度卷积神经网络看成一个黑箱，而对于其机理仍然了解甚少。同样地，深度卷积神经网络的表达能力极限是否确实比视觉词袋模型更强，也没有得到理论上的论证。

3. **视觉词袋模型和深度卷积神经网络的结合。** 视觉词袋模型和深度卷积神经网络从两种不同的角度产生图像表示，但它们本身却具有很强的联系。例如，深度卷积神经网络中的卷积计算就非常类似于视觉词袋模型中的特征聚类和编码过程。两者的竞争是否能够产生一个综合模型，其能力同时超过现有的两种模型，也将是一个非常有趣的问题。

参考文献

[1] Shannon C. A Mathematical Theory of Communication. ACM SIGMOBILE Mobile Computing and Communications Review, 2001, 5(1):3–55.

[2] Csurka G, Dance C, Fan L, et al. Visual Categorization with Bags of Keypoints. Workshop on Statistical Learning in Computer Vision, European Conference on Computer Vision, 2004, 1(22):1–2.

[3] Sivic J, Zisserman A. Video Google: A Text Retrieval Approach to Object Matching in Videos. International Conference on Computer Vision, 2003. 1470–1477.

[4] Lowe D. Distinctive Image Features from Scale-Invariant Keypoints. International Journal on Computer Vision, 2004, 60(2):91–110.

[5] Dalal N, Triggs B. Histograms of Oriented Gradients for Human Detection. Computer Vision and Pattern Recognition, 2005. 886–893.

[6] Philbin J, Chum O, Isard M, et al. Object Retrieval with Large Vocabularies and Fast Spatial Matching. Computer Vision and Pattern Recognition, 2007..

[7] Perronnin F, Sanchez J, Mensink T. Improving the Fisher Kernel for Large-scale Image Classification. European Conference on Computer Vision, 2010. 143–156.

[8] Philbin J, Chum O, Isard M, et al. Lost in Quantization: Improving Particular Object Retrieval in Large Scale Image Databases. Computer Vision and Pattern Recognition, 2008..

[9] Wang J, Yang J, Yu K, et al. Locality-Constrained Linear Coding for Image Classification. Computer Vision and Pattern Recognition, 2010. 3360–3367.

[10] Lazebnik S, Schmid C, Ponce J. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. Computer Vision and Pattern Recognition, 2006, 2:2169–2178.

[11] Krizhevsky A, Sutskever I, Hinton G. ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems, 2012. 1097–1105.

[12] Girshick R, Donahue J, Darrell T, et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. Computer Vision and Pattern Recognition, 2014. 580–587.

[13] Donahue J, Jia Y, Vinyals O, et al. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. International Conference on Machine Learning, 2014..

[14] LeCun Y, Denker J, Henderson D, et al. Handwritten Digit Recognition with a Back-Propagation Network. Advances in neural information processing systems, 1990..

[15] Dorai C, Venkatesh S. Bridging the Semantic Gap with Computational Media Aesthetics. IEEE Multimedia, 2003, 10(2):15–17.

[16] Lu Y, Zhang L, Liu J, et al. Constructing Concept Lexica With Small Semantic Gaps. IEEE Transactions on Multimedia, 2010, 12(4):288–299.

[17] Yuan J, Wu Y, Yang M. Discovery of Collocation Patterns: from Visual Words to Visual Phrases. Computer Vision and Pattern Recognition, 2007..

[18] Bosch A, Zisserman A, Muoz X. Image Classification using Random Forests and Ferns. Computer Vision and Pattern Recognition, 2007..

[19] Sanchez J, Perronnin F, Mensink T, et al. Image Classification with the Fisher Vector: Theory and Practice. International Journal of Computer Vision, 2013, 105(3):222–245.

[20] Zhang S, Tian Q, Hua G, et al. Descriptive Visual Words and Visual Phrases for Image Applications. ACM International Conference on Multimedia, 2009. 75–84.

[21] Zhang Y, Jia Z, Chen T. Image Retrieval with Geometry-Preserving Visual Phrases. Computer Vision and Pattern Recognition, 2011. 809–816.

[22] Grauman K, Darrell T. The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features. International Conference on Computer Vision, 2005, 2:1458–1465.

[23] Chum O, Philbin J, Sivic J, et al. Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval. International Conference on Computer Vision, 2007..

[24] Jing Y, Baluja S. VisualRank: Applying PageRank to Large-Scale Image Search. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008, 30(11):1877–1890.

[25] Opelt A, Fussenegger M, Pinz A, et al. Weak Hypotheses and Boosting for Generic Object Detection and Recognition. European Conference on Computer Vision, 2004. 71–84.

[26] Gradient-based learning applied to document recognition..

[27] Fergus R, Perona P, Zisserman A. Object Class Recognition by Unsupervised Scale-Invariant Learning. Computer Vision and Pattern Recognition, 2003, 2:257–264.

[28] Fei-Fei L, Fergus R, Perona P. Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories. Computer Vision and Image Understanding, 2007, 106(1):59–70.

[29] Griffin G, Holub A, Perona P. Caltech-256 Object Category Dataset. Technical Report: CNS-TR-2007-001, 2007..

[30] Deng J, Dong W, Socher R, et al. ImageNet: A Large-Scale Hierarchical Image Database. Computer Vision and Pattern Recognition, 2009. 248–255.

[31] Xiao J, Hays J, Ehinger K, et al. SUN Database: Large-Scale Scene Recognition from Abbey to Zoo. Computer Vision and Pattern Recognition, 2010. 3485–3492.

[32] Nilsback M, Zisserman A. A Visual Vocabulary for Flower Classification. Computer Vision and Pattern Recognition, 2006, 2:1447–1454.

[33] Nilsback M, Zisserman A. Automated Flower Classification over a Large Number of Classes. Indian Conference on Computer Vision, Graphics & Image Processing, 2008. 722–729.

[34] Wah C, Branson S, Welinder P, et al. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report: CNS-TR-2011-001, 2011..

[35] Khosla A, Jayadevaprakash N, Yao B, et al. Novel Dataset for Fine-Grained Image Categorization. First Workshop on Fine-Grained Visual Categorization, CVPR, 2011..

[36] Maji S, Rahtu E, Kannala J, et al. Fine-Grained Visual Classification of Aircraft. arXiv preprint, arXiv: 1306.5151, 2013..

[37] Smeulders A, Worring M, Santini S, et al. Content-based Image Retrieval at the End of the Early Years. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(12):1349–1380.

[38] Tian Q, Zhang S, Zhou W, et al. Building Descriptive and Discriminative Visual Codebook for Large-Scale Image Applications. Multimedia Tools and Applications, 2011, 51(2):441–477.

[39] Obdrzalek S, Matas J. Sub-Linear Indexing for Large Scale Object Recognition. British Machine Vision Conference, 2005..

[40] Jegou H, Douze M, Schmid C, et al. Aggregating Local Descriptors into a Compact Image Representation. Computer Vision and Pattern Recognition, 2010. 3304–3311.

[41] Felzenszwalb P, Girshick R, McAllester D, et al. Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32(9):1627–1645.

[42] Chen X, Mottaghi R, Liu X, et al. Detect What You Can: Detecting and Representing Objects Using Holistic Models and Body Parts. Computer Vision and Pattern Recognition, 2014. 1979–1986.

[43] Zheng L, Shen L, Tian L, et al. Person Re-identification Meets Image Search. arXiv preprint, arXiv: 1502.02171, 2015..

[44] Prosser B, Zheng W, Gong S, et al. Person Re-Identification by Support Vector Ranking. British Machine Vision Conference, 2010, 2(5):6.

[45] Boykov Y, Veksler O, Zabih R. Fast Approximate Energy Minimization via Graph Cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001, 23(11):1222–1239.

[46] Rother C, Kolmogorov V, Blake A. GrabCut: Interactive Foreground Extraction Using Iterated Graph Cuts. ACM Transactions on Graphics, 2004, 23(3):309–314.

[47] Davis J, Goadrich M. The Relationship between Precision-Recall and ROC Curves. International Conference on Machine Learning, 2006. 233–240.

[48] Gould S, Fulton R, Koller D. Decomposing a Scene into Geometric and Semantically Consistent Regions. International Conference on Computer Vision, 2009..

[49] Farabet C, Couprie C, Najman L, et al. Learning Hierarchical Features for Scene Labeling. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(8):1915–1929.

[50] Yamaguchi K, Kiapour M, Ortiz L, et al. Parsing Clothing in Fashion Photographs. Computer Vision and Pattern Recognition, 2012. 3570–3577.

[51] Yang W, Luo P, Lin L. Clothing Co-Parsing by Joint Image Segmentation and Labeling. Computer Vision and Pattern Recognition, 2014. 3182–3189.

[52] Martin D, Fowlkes C, Tal D, et al. A Database of Human Segmented Natural Images and Its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. International Conference on Computer Vision, 2001, 2:416–423.

[53] Arbelaez P, Maire M, Fowlkes C, et al. Contour Detection and Hierarchical Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(5):898–916.

[54] Scovanner P, Ali S, Shah M. A 3-Dimensional SIFT Descriptor and Its Application to Action Recognition. ACM International Conference on Multimedia, 2007. 357–360.

[55] Lai K, Bo L, Ren X, et al. A Large-Scale Hierarchical Multi-View RGB-D Object Dataset. International Conference on Robotics and Automation, 2011. 1817–1824.

[56] Vapnik V, Vapnik V. Statistical Learning Theory, volume 1. Wiley New York, 1998.

[57] Rosenblatt F. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. Psychological Review, 1958, 65(6):386.

[58] Minsky M, Papert S. Perceptrons - Expanded Edition: An Introduction to Computational Geometry. MIT Press Boston, 1987.

[59] Koller D, Friedman N. Probabilistic Graphical Models: Principles and Techniques. MIT Press, 2009.

[60] Bishop C. Pattern Recognition and Machine Learning. Springer New York, 2006.

[61] Mikolajczyk K, Schmid C. Scale & Affine Invariant Interest Point Detectors. International Journal of Computer Vision, 2004, 60(1):63–86.

[62] Matas J, Chum O, Urban M, et al. Robust Wide-Baseline Stereo from Maximally Stable Extremal Regions. Image and Vision Computing, 2004, 22(10):761–767.

[63] Tuytelaars T. Dense interest points. Computer Vision and Pattern Recognition, 2010. 2281–2288.

[64] Bosch A, Zisserman A, Munoz X. Scene Classification via pLSA. European Conference on Computer Vision, 2006. 517–530.

[65] Mikolajczyk K, Schmid C. A Performance Evaluation of Local Descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(10):1615–1630.

[66] Bay H, Ess A, Tuytelaars T, et al. Speeded Up Robust Features (SURF). Computer Vision and Image Understanding, 2008, 110(3):346–359.

[67] Calonder M, Lepetit V, Ozuysal M, et al. BRIEF: Computing a Local Binary Descriptor Very Fast. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(7):1281–1298.

[68] Tola E, Lepetit V, Fua P. Daisy: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32(5):815–830.

[69] Rublee E, Rabaud V, Konolige K, et al. ORB: An Efficient Alternative to SIFT or SURF. International Conference on Computer Vision, 2011. 2564–2571.

[70] Sande K, Gevers T, Snoek C. Evaluating Color Descriptors for Object and Scene Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32(9):1582–1596.

[71] Xie L, Tian Q, Wang M, et al. Spatial Pooling of Heterogeneous Features for Image Classification. IEEE Transactions on Image Processing, 2014, 23(5):1994–2008.

[72] Belongie S, Malik J, Puzicha J. Shape Matching and Object Recognition Using Shape Contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24(4):509–522.

[73] Ling H, Jacobs D. Shape Classification Using the Inner-Distance. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(2):286–299.

[74] Nister D, Stewenius H. Scalable Recognition with a Vocabulary Tree. Computer Vision and Pattern Recognition, 2006, 2:2161–2168.

[75] Jegou H, Douze M, Schmid C. Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search. European Conference on Computer Vision, 2008. 304–317.

[76] Wu C. On the convergence properties of the em algorithm. The Annals of Statistics, 1983. 95–103.

[77] Lee H, Battle A, Raina R, et al. Efficient Sparse Coding Algorithms. Advances in Neural Information Processing Systems, 2007. 801–808.

[78] Yang J, Yu K, Gong Y, et al. Linear Spatial Pyramid Matching Using Sparse Coding for Image Classification. Computer Vision and Pattern Recognition, 2009. 1794–1801.

[79] Perronnin F, Dance C. Fisher Kernels on Visual Vocabularies for Image Categorization. Computer Vision and Pattern Recognition, 2007..

[80] Jaakkola T, Haussler D, et al. Exploiting Generative Models in Discriminative Classifiers. Advances in Neural Information Processing Systems, 1999. 487–493.

[81] Zhou X, Yu K, Zhang T, et al. Image Classification using Super-Vector Coding of Local Image Descriptors. European Conference on Computer Vision, 2010. 141–154.

[82] Kobayashi T. BoF meets HOG: Feature Extraction based on Histograms of Oriented pdf Gradients for Image Classification. Computer Vision and Pattern Recognition, 2013. 747–754.

[83] Zhou W, Lu Y, Li H, et al. Scalar Quantization for Large Scale Image Search. ACM International Conference on Multimedia, 2012. 169–178.

[84] Boureau Y, Ponce J, LeCun Y. A Theoretical Analysis of Feature Pooling in Visual Recognition. International Conference on Machine Learning, 2010. 111–118.

[85] Murray N, Perronnin F. Generalized Max Pooling. Computer Vision and Pattern Recognition, 2014. 2473–2480.

[86] Feng J, Ni B, Tian Q, et al. Geometric Lp-norm Feature Pooling for Image Classification. Computer Vision and Pattern Recognition, 2011. 2609–2704.

[87] Jia Y, Huang C, Darrell T. Beyond Spatial Pyramids: Receptive Field Learning for Pooled Image Features. Computer Vision and Pattern Recognition, 2012. 3370–3377.

[88] Zhu J, Zou W, Yang X, et al. Image Classification by Hierarchical Spatial Pooling with Partial Least Squares Analysis. British Machine Vision Conference, 2012..

[89] Berg T, Belhumeur P. POOF: Part-based One-vs.-One Features for Fine-Grained Categorization, Face Verification, and Attribute Estimation. Computer Vision and Pattern Recognition, 2013. 955–962.

[90] Chai Y, Lempitsky V, Zisserman A. Symbiotic Segmentation and Part Localization for Fine-Grained Categorization. International Conference on Computer Vision, 2013. 321–328.

[91] Gavves E, Fernando B, Snoek C, et al. Fine-Grained Categorization by Alignments. International Conference on Computer Vision, 2013. 1713–1720.

[92] Liu X, Wang D, Li J, et al. The Feature and Spatial Covariant Kernel: Adding Implicit Spatial Constraints to Histogram. International Conference on Image and Video Retrieval, 2007. 565–572.

[93] Vapnik V. The Nature of Statistical Learning Theory. Springer Science & Business Media, 2000.

[94] Xie L, Tian Q, Zhang B. Feature Normalization for Part-based Image Classification. International Conference on Image Processing, 2013. 2607–2611.

[95] Akata Z, Perronnin F, Harchaoui Z, et al. Good Practice in Large-Scale Learning for Image Classification. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 36(3):507–520.

[96] Gao T, Koller D. Discriminative Learning of Relaxed Hierarchy for Large-Scale Visual Recognition. International Conference on Computer Vision, 2011. 2072–2079.

[97] Sun M, Huang W, Savarese S. Find the Best Path: an Efficient and Accurate Classifier for Image Hierarchies. International Conference on Computer Vision, 2013. 265–272.

[98] Zhang N, Farrell R, Darrell T. Pose Pooling Kernels for Sub-Category Recognition. Computer Vision and Pattern Recognition, 2012. 3665–3672.

[99] Fan R, Chang K, Hsieh C, et al. LIBLINEAR: A Library for Large Linear Classification. Journal of Machine Learning Research, 2008, 9:1871–1874.

[100] Vedaldi A, Zisserman A. Efficient Additive Kernels via Explicit Feature Maps. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(3):480–492.

[101] Jegou H, Douze M, Schmid C. On the Burstiness of Visual Elements. Computer Vision and Pattern Recognition, 2009. 1169–1176.

[102] Zheng L, Wang S, Tian Q. Lp-norm IDF for Scalable Image Retrieval. IEEE Transactions on Image Processing, 2014, 23(8):3604–3617.

[103] Kuo Y, Chen K, Chiang C, et al. Query Expansion for Hash-Based Image Object Retrieval. ACM International Conference on Multimedia, 2009. 65–74.

[104] Chum O, Mikulik A, Perdoch M, et al. Total Recall II: Query Expansion Revisited. Computer Vision and Pattern Recognition, 2011. 889–896.

[105] Chum O, Perdoch M, Matas J. Geometric Min-Hashing: Finding a (Thick) Needle in a Haystack. Computer Vision and Pattern Recognition, 2009. 17–24.

[106] Perdoch M, Chum O, Matas J. Efficient Representation of Local Geometry for Large Scale Object Retrieval. Computer Vision and Pattern Recognition, 2009. 9–16.

[107] Zhou W, Li H, Lu Y, et al. Large Scale Image Search with Geometric Coding. Proceedings of the 19th ACM international conference on Multimedia, 2011. 1349–1352.

[108] Fergus R, Perona P, Zisserman A. A Visual Category Filter for Google Images. European Conference on Computer Vision, 2004. 242–256.

[109] Kim G, Torralba A. Unsupervised Detection of Regions of Interest Using Iterative Link Analysis. Advances in Neuron Information Processing Systems, 2009. 961–969.

[110] Xie L, Tian Q, Zhou W, et al. Fast and Accurate Near-Duplicate Web Image Search with Affinity Propagation on the ImageWeb. Computer Vision and Image Understanding, 2014, 124:31–41.

[111] Turcot P, Lowe D. Better Matching with Fewer Features: The Selection of Useful Features in Large Database Recognition Problems. International Conference on Computer Vision Workshops, 2009. 2109–2116.

[112] Chum O, Matas J. Unsupervised Discovery of Co-occurrence in Sparse High Dimensional Data. Computer Vision and Pattern Recognition, 2010. 3416–3423.

[113] Jegou H, Schmid C, Harzallah H, et al. Accurate Image Search Using the Contextual Dissimilarity Measure. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32(1):2–11.

[114] Wang X, Yang M, Cour T, et al. Contextual Weighting for Vocabulary Tree based Image Retrieval. International Conference on Computer Vision, 2011. 209–216.

[115] Liu Z, Li H, Zhou W, et al. Contextual Hashing for Large-Scale Image Search. IEEE Transactions on Image Processing, 2014, 23(4):1606–1614.

[116] Qin D, Gammeter S, Bossard L, et al. Hello Neighbor: Accurate Object Retrieval with K-Reciprocal Nearest Neighbors. Computer Vision and Pattern Recognition, 2011. 777–784.

[117] Shen X, Lin Z, Brandt J, et al. Object Retrieval and Localization with Spatially-constrained Similarity Measure and k-NN Re-ranking. Computer Vision and Pattern Recognition, 2012. 3013–3020.

[118] Arandjelovic R, Zisserman A. Three Things Everyone Should Know to Improve Object Retrieval. Computer Vision and Pattern Recognition, 2012. 2911–2918.

[119] Zhang S, Yang M, Cour T, et al. Query Specific Fusion for Image Retrieval. European Conference on Computer Vision, 2012. 660–673.

[120] Qin D, Gool C. Query Adaptive Similarity for Large Scale Object Retrieval. Computer Vision and Pattern Recognition, 2013. 1610–1617.

[121] Donoser M, Bischof H. Diffusion Processes for Retrieval Revisited. Computer Vision and Pattern Recognition, 2013. 1320–1327.

[122] Fukushima K. Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. Biological Cybernetics, 1980, 36(4):193–202.

[123] Rumelhart D, Hinton G, Williams R. Learning Internal Representations by Error Propagation. Parallel Distributed Processing: Explorations in the Microstructure of Cognition, 1985..

[124] Zeiler M, Fergus R. Stochastic Pooling for Regularization of Deep Convolutional Neural Networks. arXiv preprint, arXiv: 1301.3557, 2013..

[125] He K, Zhang X, Ren S, et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. arXiv preprint, arXiv: 1406.4729, 2014..

[126] Ciresan D, Meier U, Schmidhuber J. Multi-column Deep Neural Networks for Image Classification. Computer Vision and Pattern Recognition, 2012. 3642–3649.

[127] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint, arXiv: 1409.1556, 2014..

[128] Szegedy C, Liu W, Jia Y, et al. Going Deeper with Convolutions. Advances in Neural Information Processing Systems, 2014..

[129] Hinton G, Srivastava N, Krizhevsky A, et al. Improving Neural Networks by Preventing Co-adaptation of Feature Detectors. arXiv preprint, arXiv: 1207.0580, 2012..

[130] Wan L, Zeiler M, Zhang S, et al. Regularization of Neural Networks Using Dropconnect. International Conference on Machine Learning, 2013. 1058–1066.

[131] Lee C, Xie S, Gallagher P, et al. Deeply-Supervised Nets. arXiv preprint, arXiv: 1409.5185, 2014..

[132] Lin M, Chen Q, Yan S. Network in Network. arXiv preprint, arXiv: 1312.4400, 2013..

[133] Goodfellow I, Warde-Farley D, Mirza M, et al. Maxout Networks. International Conference on Machine Learning, 2013. 1319–1327.

[134] Zheng L, Wang S, He F, et al. Seeing the Big Picture: Deep Embedding with Contextual Evidences. arXiv preprint, arXiv: 1406.0132, 2014..

[135] Wu Z, Huang Y, Yu Y, et al. Early Hierarchical Contexts Learned by Convolutional Networks for Image Segmentation. International Conference on Pattern Recognition, 2014. 1538–1543.

[136] Mao J, Xu W, Yang Y, et al. Deep Captioning with Multimodal Recurrent Neural Networks (m-RNN). arXiv preprint, arXiv: 1412.6632, 2014..

[137] Abdel-Hamid O, Mohamed A, Jiang H, et al. Applying Convolutional Neural Networks Concepts to Hybrid NN-HMM Model for Speech Recognition. 2012. 4277–4280.

[138] Jia Y, Shelhamer E, Donahue J, et al. CAFFE: Convolutional Architecture for Fast Feature Embedding. ACM International Conference on Multimedia, 2014. 675–678.

[139] Pan S, Yang Q. A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(10):1345–1359.

[140] Chatfield K, Simonyan K, Vedaldi A, et al. Return of the Devil in the Details: Delving Deep into Convolutional Nets. British Machine Vision Conference, 2014..

[141] Razavian A, Azizpour H, Sullivan J, et al. CNN Features off-the-shelf: an Astounding Baseline for Recognition. Computer Vision and Pattern Recognition, 2014. 512–519.

[142] Liang X, Liu S, Shen X, et al. Deep Human Parsing with Active Template Regression. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, PP(99).

[143] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space. Advances in Neural Information Processing Systems, 2013..

[144] Liu J, Sun J, Shum H. Paint Selection. ACM Transactions on Graphics, 2009, 28(3):69.

[145] Canny J. A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986, (6):679–698.

[146] Kanopoulos N, Vasanthavada N, Baker R. Design of an Image Edge Detection Filter Using the Sobel Operator. IEEE Journal of Solid-State Circuits, 1988, 23(2):358–367.

[147] Haralick R. Digital Step Edges from Zero Crossing of Second Directional Derivatives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1984, (1):58–68.

[148] Eklundh J, Elfving T, Nyberg S. Edge Detection Using the Marr-Hildreth Operator with Different Sizes. International Conference on Pattern Recognition, 1982, 6:1109–1112.

[149] Ruzon M, Tomasi C. Color Edge Detection with the Compass Operator. Computer Vision and Pattern Recognition, 1999, 2.

[150] Hwang J, Liu T. Pixel-wise Deep Learning for Contour Detection. arXiv preprint, arXiv: 1504.01989, 2015..

[151] Liu C, Wechsler H. Gabor Feature Based Classification Using the Enhanced Fisher Linear Discriminant Model for Face Recognition. IEEE Transactions on Image processing, 2002, 11(4):467–476.

[152] Viola P, Jones M. Rapid Object Detection Using a Boosted Cascade of Simple Features. Computer Vision and Pattern Recognition, 2001. 504–511.

[153] Felzenszwalb P, Girshick R, McAllester D. Cascade Object Detection with Deformable Part Models. Computer Vision and Pattern Recognition, 2010. 2241–2248.

[154] Wang X, Yang M, Zhu S, et al. Regionlets for Generic Object Detection. International Conference on Computer Vision, 2013. 17–24.

[155] Vondrick C, Khosla A, Malisiewicz T, et al. HOGgles: Visualizing Object Detection Features. International Conference on Computer Vision, 2013. 1–8.

[156] Alexe B, Deselaers T, Ferrari V. Measuring the Objectness of Image Windows. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(11):2189–2202.

[157] Uijlings J, Sande K, Gevers T, et al. Selective Search for Object Recognition. International journal of computer vision, 2013, 104(2):154–171.

[158] Cheng M, Zhang Z, Lin W, et al. BING: BInarized Normed Gradients for Objectness Estimation at 300fps. Computer Vision and Pattern Recognition, 2014. 3286–3293.

[159] Zhang N, Donahue J, Girshick R, et al. Part-based R-CNNs for Fine-Grained Category Detection. European Conference on Computer Vision, 2014. 834–849.

[160] Jegou H, Tavenard R, Douze M, et al. Searching in One Billion Vectors: Re-rank with Source Coding. International Conference on Acoustics, Speech and Signal Processing, 2011. 861–864.

[161] Indyk P, Motwani R. Approximate nearest neighbors: towards removing the curse of dimensionality. ACM Annual Symposium on Theory of Computing, 1998. 604–613.

[162] Arya S, Mount D. Approximate Nearest Neighbor Queries in Fixed Dimensions. 1993, 93:271–280.

[163] Arya S, Mount D, Netanyahu N, et al. An Optimal Algorithm for Approximate Nearest Neighbor Searching Fixed Dimensions. Journal of the ACM, 1998, 45(6):891–923.

[164] Silpa-Anan C, Hartley R. Optimised KD-trees for Fast Image Descriptor Matching. Computer Vision and Pattern Recognition, 2008..

[165] Muja M, Lowe D. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. International Conference on Computer Vision Theory and Application, 2009, 2:331–340.

[166] Norouzi M, Blei D. Minimal Loss Hashing for Compact Binary Codes. International Conference on Machine Learning, 2011. 353–360.

[167] Datar M, Immorlica N, Indyk P, et al. Locality-Sensitive Hashing Scheme Based on p-Stable Distributions. Annual Symposium on Computational Geometry, 2004. 253–262.

[168] Jegou H, Douze M, Schmid C. Product Quantization for Nearest Neighbor Search. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(1):117–128.

[169] Ge T, He K, Ke Q, et al. Optimized Product Quantization for Approximate Nearest Neighbor Search. Computer Vision and Pattern Recognition, 2013. 2946–2953.

[170] Zhang T, Du C, Wang J. Composite Quantization for Approximate Nearest Neighbor Search. International Conference on Machine Learning, 2014. 838–846.

[171] Chatfield K, Lempitsky V, Vedaldi A, et al. The Devil is in the Details: An Evaluation of Recent Feature Encoding Methods. British Machine Vision Conference, 2011..

[172] Xie L, Tian Q, Zhang B. Max-SIFT: Flipping Invariant Descriptors for Web Logo Search. International Conference on Image Processing, 2014. 5716–5720.

[173] Xie L, Tian Q, Wang J, et al. Image Classification with Max-SIFT Descriptors. International Conference on Acoustics, Speech and Signal Processing, 2015..

[174] Vedaldi A, Fulkerson B. VLFeat: An Open and Portable Library of Computer Vision Algorithms. ACM Multimedia, 2010. 1469–1472.

[175] Paulin M, Revaud J, Harchaoui Z, et al. Transformation Pursuit for Image Classification. Computer Vision and Pattern Recognition, 2014. 3646–3653.

[176] Guo X, Cao X. FIND: A Neat Flip Invariant Descriptor. International Conference on Pattern Recognition, 2010. 515–518.

[177] Ma R, Chen J, Su Z. MI-SIFT: Mirror and Inversion Invariant Generalization for SIFT Descriptor. International Conference on Image and Video Retrieval, 2010. 228–235.

[178] Zhao W, Ngo C. Flip-Invariant SIFT for Copy and Object Detection. IEEE Transactions on Image Processing, 2013, 22(3):980–991.

[179] Parkhi O, Vedaldi A, Zisserman A, et al. Cats and Dogs. Computer Vision and Pattern Recognition, 2012. 3498–3505.

[180] Lapin M, Schiele B, Hein M. Scalable Multitask Representation Learning for Scene Classification. Computer Vision and Pattern Recognition, 2014. 1434–1441.

[181] Angelova A, Zhu S. Efficient Object Detection and Segmentation for Fine-Grained Recognition. Computer Vision and Pattern Recognition, 2013. 811–818.

[182] Pu J, Jiang Y, Wang J, et al. Which Looks Like Which: Exploring Inter-class Relationships in Fine-Grained Visual Categorization. European Conference on Computer Vision, 2014. 425–440.

[183] Wang Z, Feng J, Yan S. Collaborative Linear Coding for Robust Image Classification. International Journal of Computer Vision, 2014, (1):1–12.

[184] Zhang N, Farrell R, Iandola F, et al. Deformable Part Descriptors for Fine-Grained Recognition and Attribute Prediction. International Conference on Computer Vision, 2013. 729–736.

[185] Juneja M, Vedaldi A, Jawahar C, et al. Blocks that Shout: Distinctive Parts for Scene Classification. Computer Vision and Pattern Recognition, 2013. 923–930.

[186] Kobayashi T. Dirichlet-based Histogram Feature Transform for Image Classification. Computer Vision and Pattern Recognition, 2014. 3278–3285.

[187] Xie L, Wang J, Guo B, et al. Orientational Pyramid Matching for Recognizing Indoor Scenes. Computer Vision and Pattern Recognition, 2014. 3734–3741.

[188] Yang Y, Newsam S. Bag-of-Visual-Words and Spatial Extensions for Land-Use Classification. International Conference on Advances in Geographic Information Systems, 2010. 270–279.

[189] Quattoni A, Torralba A. Recognizing Indoor Scenes. Computer Vision and Pattern Recognition, 2009. 413–420.

[190] Boureau Y, Bach F, LeCun Y, et al. Learning Mid-Level Features for Recognition. Computer Vision and Pattern Recognition, 2010. 2559–2566.

[191] Xie L, Tian Q, Zhang B. Spatial Pooling of Heterogeneous Features for Image Applications. ACM International Conference on Multimedia, 2012. 539–548.

[192] Xie L, Tian Q, Hong R, et al. Hierarchical Part Matching for Fine-Grained Visual Categorization. International Conference on Computer Vision, 2013. 1641–1648.

[193] Itti L, Koch C, Niebur E. A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, 20(11):1254–1259.

[194] Everingham M, Van Gool L, Williams C K I, et al. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. 2007..

[195] Gao S, Tsang I, Chia L. Kernel Sparse Representation for Image Classification and Face Recognition. European Conference on Computer Vision, 2010. 1–14.

[196] Lazebnik S, Schmid C, Ponce J, et al. Semi-Local Affine Parts for Object Recognition. British Machine Vision Conference, 2004. 779–788.

[197] Larlus D, Jurie F. Latent Mixture Vocabularies for Object Categorization and Segmentation. Image and Vision Computing, 2009, 27(5):523–534.

[198] Gehler P, Nowozin S. On Feature Combination for Multiclass Object Classification. International Conference on Computer Vision, 2009. 221–228.

[199] Li L, Su H, Fei-Fei L, et al. Object Bank: A High-Level Image Representation for Scene Classification & Semantic Feature Sparsification. Advances in neural information processing systems, 2010. 1378–1386.

[200] Bo L, Ren X, Fox D. Hierarchical Matching Pursuit for Image Classification: Architecture and Fast Algorithms. Advances in Neural Information Processing Systems, 2011. 2115–2123.

[201] Xie L, Tian Q, Zhang B. Generalized Regular Spatial Pooling for Image Classification. International Conference on Acoustics, Speech and Signal Processing, 2015..

[202] Li L, Fei-Fei L. What, Where and Who? Classifying Events by Scene and Object Recognition. International Conference on Computer Vision, 2007. 1–8.

[203] Bo L, Ren X, Fox D. Multipath Sparse Coding Using Hierarchical Matching Pursuit. Computer Vision and Pattern Recognition, 2013. 660–667.

[204] Arbelaez P, Maire M, Fowlkes C, et al. From Contours to Regions: An Empirical Evaluation. Computer Vision and Pattern Recognition, 2009. 2294–2301.

[205] Chen Q, Song Z, Hua Y, et al. Hierarchical Matching with Side Information for Image Classification. Computer Vision and Pattern Recognition, 2012. 3426–3433.

[206] Yang S, Bo L, Wang J, et al. Unsupervised Template Learning for Fine-Grained Object Recognition. Advances in Neural Information Processing Systems, 2012. 3122–3130.

[207] Haines O, Calway A. Detecting Planes and Estimating their Orientation from a Single Image. British Machine Vision Conference, 2012. 1–11.

[208] Xie L, Tian Q, Zhou W, et al. Heterogeneous Graph Propagation for Large-Scale Web Image Search. IEEE Transactions on Image Processing, 2015, PP(99).

[209] Kleinberg J. Authoritative Sources in a Hyperlinked Environment. Journal of the ACM, 1999, 46(5):604–632.

[210] Brin S, Page L. The Anatomy of a Large-Scale Hypertextual Web Search Engine. Computer Networks and ISDN Systems, 1998, 30(1):107–117.

[211] Tolias G, Jegou H. Visual Query Expansion with or without Geometry: Refining Local Descriptors by Feature Aggregation. Pattern Recognition, 2014, 47(10):3466–3476.

[212] Deng C, Ji R, Liu W, et al. Visual Reranking through Weakly Supervised Multi-Graph Learning..

[213] Ma H, Zhu J, Lyu M, et al. Bridging the Semantic Gap between Image Contents and Tags. IEEE Transactions on Multimedia, 2010, 12(5):462–473.

[214] Mikulik A, Perdoch M, Chum O, et al. Learning Vocabularies over a Fine Quantization. International Journal of Computer Vision, 2013, 103(1):163–175.

[215] Frey B, Dueck D. Clustering by Passing Messages between Data Points. Science, 2007, 315(5814):972–976.

[216] Buckland M, Gey F. The Relationship between Recall and Precision. Journal of the American Society for Information Science, 1994, 45(1):12–19.

[217] Boiman O, Shechtman E, Irani M. In Defense of Nearest-Neighbor Based Image Classification. Computer Vision and Pattern Recognition, 2008. 1–8.

[218] Xie L, Hong R, Zhang B, et al. Image Classification and Retrieval are ONE. International Conference on Multimedia Retrieval, 2015..

[219] Branson S, Van Horn G, Belongie S, et al. Bird Species Categorization Using Pose Normalized Deep Convolutional Nets. arXiv preprint, arXiv: 1406.2952, 2014..

[220] Zheng L, Wang S, Liu Z, et al. Packing and Padding: Coupled Multi-Index for Accurate Image Retrieval. Computer Vision and Pattern Recognition, 2014. 1947–1954.

[221] Zhang S, Yang M, Wang X, et al. Semantic-aware Co-indexing for Image Retrieval. International Conference on Computer Vision, 2013. 1673–1680.

[222] Xie L, Wang J, Zhang B, et al. Fine-Grained Image Search. IEEE Transactions on Multimedia, 2015..

[223] Xie L, Lin W, Xu Y, et al. Webpage Popularity vs. Aesthetics: A Case Study Based on Visual Content Analysis. *submitted, to* ACM International Conference on Multimedia, 2015..

[224] Romberg S, Pueyo L, Lienhart R, et al. Scalable Logo Recognition in Real-World Images. International Conference on Multimedia Retrieval, 2011. 25–32.

[225] Fellbaum C. WordNet. Wiley Online Library, 1998.

[226] Bossard L, Guillaumin M, Gool L. Food-101 – Mining Discriminative Components with Random Forests. European Conference on Computer Vision, 2014. 446–461.

[227] Jarvelin K, Kekalainen J. Cumulated Gain-based Evaluation of IR Techniques. ACM Transactions on Information Systems, 2002, 20(4):422–446.

[228] Wang Y, Liwei W, Li Y, et al. A Theoretical Analysis of nDCG Ranking Measures. Annual Conference on Learning Theory, 2013..

[229] Parikh D, Grauman K. Relative Attributes. International Conference on Computer Vision, 2011. 503–510.

[230] Deng J, Berg A, Fei-Fei L. Hierarchical Semantic Indexing for Large Scale Image Retrieval. Computer Vision and Pattern Recognition, 2011. 785–792.

[231] Wang J, Leung T, Rosenberg C, et al. Learning Fine-Grained Image Similarity with Deep Ranking. Computer Vision and Pattern Recognition, 2014. 1386–1393.

[232] Cai D, Yu S, Wen J, et al. VIPS: A Vision-based Page Segmentation Algorithm. Technical report, Microsoft Technical Report, MSR-TR-2003-79, 2003.

[233] De Angeli A, Sutcliffe A, Hartmann J. Interaction, Usability and Aesthetics: What Influences Users' Preferences? Conference on Designing Interactive Systems, 2006. 271–280.

[234] Schmidt K, Liu Y, Sridharan S. Webpage Aesthetics, Performance and Usability: Design Variables and Their Effects. Ergonomics, 2009, 52(6):631–643.

[235] Oliva A, Torralba A. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. International Journal of Computer Vision, 2001, 42(3):145–175.

[236] Chapelle O, Zhang Y. A Dynamic Bayesian Network Click Model for Web Search Ranking. International Conference on World Wide Web, 2009. 1–10.

[237] Guo F, Liu C, Wang Y. Efficient Multiple-Click Models in Web Search. International Conference on Web Search and Data Mining, 2009. 124–131.

[238] Ren X, Wang Y, Yu X, et al. Heterogeneous Graph-Based Intent Learning with Queries, Web Pages and Wikipedia Concepts. International Conference on Web Search and Data Mining, 2014. 23–32.

[239] Wang X, Jia J, Hu P, et al. Understanding the Emotional Impact of Images. ACM International Conference on Multimedia, 2012. 1369–1370.

[240] Papachristos E, Avouris N. The Subjective and Objective Nature of Website Aesthetic Impressions. Human-Computer Interaction, 2009. 119–122.

[241] Schenkman B, Jonsson F. Aesthetics and Preferences of Web Pages. Behaviour & Information Technology, 2000, 19(5):367–377.

[242] Cheng B, Ni B, Yan S, et al. Learning to Photograph. ACM International Conference on Multimedia, 2010. 291–300.

[243] Glommen C, Barrelet B. Internet Website Traffic Flow Analysis, 2002. US Patent 6,393,479.

[244] Khosla A, Das Sarma A, Hamid R. What Makes an Image Popular? International Conference on World Wide Web, 2014. 867–876.

[245] Lu X, Lin Z, Jin H, et al. RAPID: Rating Pictorial Aesthetics using Deep Learning. ACM Multimedia, 2014. 457–466.

[246] Shimamura A, Palmer S. Aesthetic Science: Connecting Minds, Brains, and Experience. Oxford University Press, 2012.

[247] Brown S, Gao X, Tisdelle L, et al. Naturalizing Aesthetics: Brain Areas for Aesthetic Appraisal across Sensory Modalities. Neuroimage, 2011, 58(1):250–258.

致　谢

五年的博士研究生涯是一次漫长的旅途。我很庆幸自己能够最终坚持走完这段路。在这个过程中，我得到了很多人的帮助，在此一并致谢。

我要感谢我的导师张钹院士对我的指导。张老师的言传身教将使我受益终身。在我读博期间，张老师为我提供了自由宽松的学习工作环境，让我能够不受外界干扰，潜心科研。同时，张老师还为我提供了大量的合作交流机会，让我能够开拓视野，增长更多见识。

我要感谢我的副导师田奇教授。田老师在我科研的起步阶段，对我进行了耐心细致的指导，使我逐步走上研究的正轨。我在美国德克萨斯大学圣安东尼奥分校访学期间，田老师也对我的生活给予了充分的照顾。

我要感谢我在微软亚洲研究院的导师王井东研究员。他不仅为我提供了宝贵的实习机会，还从一个科研前辈的角度，指导并改进了我的科研方式。

我要感谢智能技术与系统国家重点实验室的李建民老师、胡晓林老师和朱军老师。三位老师不仅对我的科研进行了耐心的指导，还给予了我生活上的关心和照顾。

我要感谢智能技术与系统国家重点实验室的袁进辉、左圆圆、刘啸冰和刘才良几位师兄和师姐。他们传授给我的经验，帮助我度过了彷徨的科研起步阶段。

我要感谢智能技术与系统国家重点实验室的同学们，包括：梁鸣、张晓露、季剑秋、徐旻捷、周以苏、张清天、陈蓓、李巍、吴裔慧、张傲南、郭金马、胡文波、田天，等等。我们之间的讨论和合作卓有成效，聚餐和娱乐也将成为我难忘的回忆。

我要感谢我的父母对我读博的支持和理解，也要感谢我的女友始终与我相伴。你们的陪伴让我心安，鼓励我走过最困难的阶段。

我要感谢所有帮助过我，鼓励过我的人。你们的关心一直给予我信心和力量。

我也要感谢那些轻视我的人。你们的质疑也是激励我不断奋进的动力。

声　明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签　名：_____ 日　期：_____

附录 A　RIDE算法的补充说明

本附录作为第3章RIDE算法的补充说明，将提供一个关于SIFT梯度估计的证明，以及一个将RIDE算法推广到其他翻转不变性情形的理论和实验分析。

A.1　密集SIFT特征的朝向估计

本节的目标是给出密集采样SIFT特征的朝向的近似估计。这种估计方式完全依赖于SIFT本身的128维梯度强度值，可以用于快速计算朝向向量（见第3.3.3节）。

A.1.1　SIFT的实现

SIFT的实现方式主要依赖于原始论文[4]。在接下来的段落中，我们简单地概述一下其中的朝向分配和特征表示过程。

首先，我们假设特征的尺度已经计算完成：这符合密集采样的特性，此时所有的图像块具有完全相同的大小。将一张图像表示为$\mathbf{I} = [L(x,y)]_{W\times H}$。每个像素点的局部梯度的大小$m(x,y)$和方向$\theta(x,y)$已经事先计算完毕：

$$m(x,y) = \left[(L(x+1,y)-L(x-1,y))^2 + (L(x,y+1)-L(x,y-1))^2\right]^{1/2} \quad \text{(A-1)}$$

$$\theta(x,y) = \arctan\left[(L(x,y+1)-L(x,y-1)) / (L(x+1,y)-L(x-1,y))\right] \quad \text{(A-2)}$$

像素点的梯度大小和方向随后被用于估计整个描述子的*主方向*（dominant orientation）。为此，我们通过这个局部区块内的所有像素的朝向，建立一个朝向直方图。每个样本的梯度，经过高斯窗函数（方差σ为图像区块大小的1.5倍）加权后，被累加到梯度直方图里。直方图里的峰值，就对应着整个描述子的主方向。我们找到直方图内的最高峰值，以及那些峰值高于最高值80%的位置，在其相应方向上建立一个描述子。这样，在那些具有多个相近峰值的位置，可能会出现多个不同的描述子，它们具有相同的位置和不同的方向。这种朝向角度计算方法称为**主方向法**。

上述的方法在图像匹配和检索[4]问题中能够很好地工作，但是在图像分类任务中，我们通常并不需要在一个位置上出现多个不同的描述子。此时，我们也可以通过如下方法，估计出唯一确定的朝向。首先，每个像素上的梯度在x和y方向

上被分解，即：

$$m_x(x,y) = m(x,y) \times \cos\theta(x,y) \tag{A-3}$$

$$m_y(x,y) = m(x,y) \times \sin\theta(x,y) \tag{A-4}$$

同时，分解后的分量在x和y方向上被相应地累加起来：

$$G_x(x,y) = \sum_{x,y} m_x(x,y) \tag{A-5}$$

$$G_y(x,y) = \sum_{x,y} m_y(x,y) \tag{A-6}$$

这样我们就得到了一个二维向量$\mathbf{G}$作为这个SIFT描述子的朝向。这种朝向角度计算方法称为**累加法**。

当然，我们也可以遵循原始的实现方式[4]为每个密集采样的特征分配朝向。在实际操作中，我们实现了上述两种方式（主方向法和累加法），并且发现后者的分类效果稍好于前者。我们偏好累加法的另外一个原因是，它产生的朝向是$[0, 2\pi)$之间的连续值，这也使得RIDE-8算法的效果更加稳定（见第A.2.1节）。

关于特征描述，我们利用每个像素处的$m(x,y)$和$\theta(x,y)$值来完成。普通SIFT特征[4]需要将图像块按照主方向进行对齐（旋转使得主方向朝上），而密集SIFT特征[174]的实现不需要旋转图像块。对齐后的图像块被分为4×4的网格，每个网格内则需要计算一个8维的向量，表示8个朝向上的梯度强度信息。每个像素的梯度强度值$m(x,y)$将被按照线性插值的方式分配到至多2个不同的维度上，假设某个像素的方向为$\theta(x,y)$，首先找到其最近邻的两个梯度方向$\theta_a < \theta(x,y) < \theta_b$，然后通过下式计算插值的权值：

$$m_a = m(x,y) \times \frac{\theta_b - \theta(x,y)}{\theta_b - \theta_a} \tag{A-7}$$

$$m_b = m(x,y) \times \frac{\theta(x,y) - \theta_a}{\theta_b - \theta_a} \tag{A-8}$$

这样，在4×4切分中的每个网格内，我们都得到了一个8维的特征向量。将所有16个特征向量拼接起来后，就能够得到一个总体的128维描述子向量。

A.1.2 重构SIFT的整体朝向

我们希望，即使对于密集采样的SIFT（不具有主方向信息），我们也能利用局部的梯度强度值，重构SIFT的主方向。重构的方法已经在第3.3.3节中叙述。这一节的主要目的是证明下述定理：

定理 A.1： 给定一个密集采样的SIFT特征$\mathbf{d} = (d_k, \theta_k)_{k=0,1,\ldots,127}$，其中$d_k$和$\theta_k$代表其中第$k$维的梯度强度值和对应的方向角。这个特征的**累加法**主方向θ近似满足下式：

$$\tan\theta = \frac{G_y(x,y)}{G_x(x,y)} = \frac{\sum_{x,y} m_y(x,y)}{\sum_{x,y} m_x(x,y)} \approx \frac{\sum_k d_k \sin\theta_k}{\sum_k d_k \cos\theta_k} \tag{A-9}$$

为此，我们只需要证明如下引理：

引理 A.1： 当一个具有任意方向的梯度强度值(m,θ)通过线性插值被量化为(m_a,θ_a)和(m_b,θ_b)（$\theta_a < \theta < \theta_b$），即：

$$m_a = m \times \frac{\theta_b - \theta}{\theta_b - \theta_a} \tag{A-10}$$

$$m_b = m \times \frac{\theta - \theta_a}{\theta_b - \theta_a} \tag{A-11}$$

量化前后的强度值在分解后的效果是近似相同的，即：

$$m\cos\theta \approx m_a\cos\theta_a + m_b\cos\theta_b \tag{A-12}$$

$$m\sin\theta \approx m_a\sin\theta_a + m_b\sin\theta_b \tag{A-13}$$

证明 我们只证明(A-12)因为(A-13)的证明与其是非常类似的。利用式(A-10)和(A-11)代替式(A-12)中的m_a和m_b，得到：

$$\begin{aligned} & m_a\cos\theta_a + m_b\cos\theta_b \\ = \; & m \times \frac{\theta_b - \theta}{\theta_b - \theta_a} \times \cos\theta_a + m \times \frac{\theta - \theta_a}{\theta_b - \theta_a} \times \cos\theta_b && \text{(A-14)} \\ = \; & m \times \left(\frac{\theta_b - \theta}{\theta_b - \theta_a} \times \cos\theta_a + \frac{\theta - \theta_a}{\theta_b - \theta_a} \times \cos\theta_b \right) && \text{(A-15)} \end{aligned}$$

现在，我们利用如下近似：

$$\frac{\theta_b - \theta}{\theta_b - \theta_a} \approx \frac{\sin(\theta_b - \theta)}{\sin(\theta_b - \theta_a)} \tag{A-16}$$

$$\frac{\theta - \theta_a}{\theta_b - \theta_a} \approx \frac{\sin(\theta - \theta_a)}{\sin(\theta_b - \theta_a)} \tag{A-17}$$

于是(A-15)式成为：

$$\begin{aligned} & m_a\cos\theta_a + m_b\cos\theta_b \\ = \; & m \times \frac{\theta_b - \theta}{\theta_b - \theta_a} \times \cos\theta_a + m \times \frac{\theta - \theta_a}{\theta_b - \theta_a} \times \cos\theta_b && \text{(A-18)} \\ \approx \; & m \times \left[\frac{\sin(\theta_b - \theta)}{\sin(\theta_b - \theta_a)} \times \cos\theta_a + \frac{\sin(\theta - \theta_a)}{\sin(\theta_b - \theta_a)} \times \cos\theta_b \right] && \text{(A-19)} \\ = \; & \frac{m \times [\sin(\theta_b - \theta)\cos\theta_a + \sin(\theta - \theta_a)\cos\theta_b]}{\sin(\theta_b - \theta_a)} && \text{(A-20)} \end{aligned}$$

$$= \frac{m \times [(\sin\theta_b \cos\theta - \cos\theta_b \sin\theta)\cos\theta_a + (\sin\theta\cos\theta_a - \cos\theta\sin\theta_a)\cos\theta_b]}{\sin(\theta_b - \theta_a)} \tag{A-21}$$

$$= \frac{m \times [\cos\theta(\sin\theta_b \cos\theta_a - \cos\theta_b \sin\theta_a)]}{\sin(\theta_b - \theta_a)} \tag{A-22}$$

$$= m\cos\theta \tag{A-23}$$

我们对式(A-16)和(A-17)进行一个简单的讨论。由于确定了$\theta_b - \theta_a = \pi/4$，上述近似的最大相对误差小于11%。为了进一步分析，首先定义$f(x) = \frac{\sin x}{x}$。由于$\lim_{x\to 0} f(x) = 1$，且$f(x)$是一个单调增加的函数。式(A-16)和(A-17)的最大误差出现在$\theta_b - \theta$或者$\theta - \theta_a$非常小的时候。在这种情况下，m_a或者m_b也非常小，因此近似造成的绝对误差也非常小，几乎可以忽略不计。因此，我们认为式(A-16)和(A-17)的近似是可行的。 □

A.2 RIDE的推广：RIDE-4和RIDE-8

本节中，我们将讨论RIDE算法在更多翻转和旋转不变性上的推广。这是正文第3.3.4节的扩展部分。

A.2.1 RIDE-2、RIDE-4和RIDE-8

首先回顾我们曾经计算了一个二维向量$\mathbf{G} = \left(G_x, G_y\right)^\top$用于表达SIFT特征的朝向信息，其中$G_x$和$G_y$分别近似表示了SIFT特征朝右以及朝下的程度。如果限制$G_x \geqslant 0$对于任何描述子$\mathbf{d}$都成立，那么我们就需要针对$\mathbf{d}$生成一个左右翻转的版本，并且从两个候选$\mathbf{d}$和$\mathbf{d}^{\mathrm{R}}$中找到满足$G_x \geqslant 0$条件的一个。这个描述子，记为$r_2(\mathbf{d})$，显然满足左右翻转不变特性。当然，如果对于$\mathbf{d}$而言有$G_x = 0$，那么翻转前后的版本都能够满足翻转不变的要求。在这种情况下，我们寻找那个具有较大字典序的候选者：这一技巧也可以应用于后续的算法中，即**RIDE-4**和**RIDE-8**。

如果我们同时还需要达到上下翻转不变的特性，那么相应地，G_y也需要被限制为$G_y \geqslant 0$。于是，我们生成其他三个$\mathbf{d}$的翻转版本，即$\mathbf{d}_0$、$\mathbf{d}_1$、$\mathbf{d}_2$和$\mathbf{d}_3$，其中$\mathbf{d}_0$就是$\mathbf{d}$，$\mathbf{d}_1$是$\mathbf{d}$经过左右翻转后的版本，$\mathbf{d}_2$是$\mathbf{d}$经过上下翻转后的版本，而$\mathbf{d}_3$是$\mathbf{d}$同时左右和上下翻转后的版本。显然，这四个候选中至少有一个同时满足$G_x \geqslant 0$和$G_y \geqslant 0$。这一特征，记为$r_4(\mathbf{d})$，同时满足左右和上下翻转不变的特性。

最后一种情形（**RIDE-8**）能够处理将特征旋转90°后的不变性。在上述变换（左右翻转、上下翻转）中追加旋转90°，将得到总共八个不同的版本。我们生成所有这些变种，并且在其中选择满足$G_x \geqslant G_y \geqslant 0$的一个。将这个条件展开，能够

算法	**Aircraft-100**-1	**Aircraft-100**-2	**Aircraft-100**-4	**Aircraft-100**-8
RIDE	**58.75**	48.52	39.33	25.11
RIDE-2	55.22	**55.22**	43.20	29.71
RIDE-4	47.44	47.44	**47.44**	35.41
RIDE-8	43.47	43.47	43.47	**43.47**

表 A.1 不同版本的RIDE算法在不同版本的**Aircraft-100**数据集上的分类精度(%)。

得到$G_x \geqslant 0$、$G_y \geqslant 0$和$G_x \geqslant G_y$。这一特征，记为$r_8(\mathbf{d})$，同时满足左右、上下翻转不变以及90°旋转不变的特性。

我们提供一种针对**RIDE-2**、**RIDE-4**和**RIDE-8**算法的直观性解释。所有这些翻转和旋转的操作，都会相应地改变描述子的主方向。**RIDE-2**通过确定$G_x \geqslant 0$将描述子的主方向限制在180°的范围内，这个范围在**RIDE-4**中被压缩至90°，而在**RIDE-8**中被进一步压缩至45°。一个具有任意朝向的描述子都可以通过翻转和旋转操作达到这样的要求，而在此过程中我们抵消了可能出现的翻转和旋转操作，从而产生了需要的翻转和旋转不变性。

A.2.2 实验

我们在**Aircraft-100**数据集[36]上测试原始SIFT特征和它在RIDE-2、RIDE-4以及RIDE-8下的变种。我们使用四个**Aircraft-100**数据集的不同变种。第一个对齐的版本，记为**Aircraft-100**-1，是将所有原始图像的朝向都翻转至朝右后获得的。其他三个版本，分别记为**Aircraft-100**-2、**Aircraft-100**-4和**Aircraft-100**-8，分别将对齐版本里的每张图像随机替换为2个、4个和8个经过翻转或旋转后的变种。这里，2个变换指的是不变和左右翻转；4个变换是在2个变换的基础上加入上下翻转；而8个变换则是在4个变换的基础上加入90°旋转。其中，**Aircraft-100**-2版本与原先的**Aircraft-100**数据集（没有经过对齐）非常类似。

基本设置与正文第3.4.1节中设置相同。在本节的实验中，我们只用SIFT特征，并且不使用空间金字塔：这样简单的模型已经足以说明问题。分类的结果总结在表A.1中。我们可以观察到，在**Aircraft-100**-1数据集上，利用原始SIFT特征的模型（**ORIG**）产生了最好的分类结果。原始特征被任何一种RIDE算法处理过后，分类的精度都显著地下降了。潜在的原因是RIDE算法通过一对多的挑选，损害了局部特征的描述能力：挑选的范围越大，精度的下降就越明显。

在**Aircraft-100**-2数据集上，**RIDE-2**算法比**ORIG**取得了更好的效果。这就意味着**RIDE-2**算法通过捕捉左右翻转不变性，达到了提升精度的目的。尽管原始SIFT特征的描述能力被减弱了，但是我们从翻转不变中获得的利益更大，超过

了描述力减弱带来的负面作用。然而，应用**RIDE-4**和**RIDE-8**算法时，描述子的描述能力继续下降，且我们并不能从增强的不变性上获得更多的好处（因为没有相应的图像样例），这就导致了**RIDE-4**和**RIDE-8**的分类精度不及**RIDE-2**。类似的结果还在**Aircraft-100**-4数据集上观测到：此时**RIDE-4**的分类精度超过了**RIDE-2**，但是**RIDE-8**的分类精度不如**RIDE-4**。最后，在**Aircraft-100**-8数据集上，所有的翻转和旋转不变性都起了作用，因此**RIDE-8**取得了最好的分类效果。

上述实验告诉我们，RIDE算法产生的作用有两个：为局部描述子提供某种不变性，并且降低描述子的描述能力。根据表A.1的结果估计，当描述子能够处理一种必要的不变性时，我们能够得到大约10%的精度提升；同时，多处理一种不变性所带来的描述能力的损失，会造成大约5%的精度下降。因此，最优的策略并不是覆盖所有的不变性：**最好的方法是只覆盖那些在数据集中出现的不变性。**

相应地，我们在正文的实验中不使用**RIDE-4**和**RIDE-8**算法，因为所有的数据集都没有上下翻转或者旋转的样例出现。在这些细粒度分类和场景分类问题中，通常只会出现左右翻转的图像样例，因此**RIDE-2**始终是最佳选择。

个人简历、在学期间发表的学术论文与研究成果

个人简历

1988年8月1日出生于福建省福州市。

2006年9月保送进入清华大学理学院数理基础科学专业， 2008年7月转系进入清华大学计算机科学与技术系计算机科学与技术专业， 2010年7月本科毕业并且获得工学学士学位。

2010年9月免试进入清华大学计算机科学与技术系计算机科学与技术专业攻读工学博士学位至今。

在学期间发表的学术论文

[1] **Xie L**, Tian Q, Zhang B. Simple Techniques Make Sense: Feature Pooling and Normalization for Image Classification. *to appear*, in IEEE Transaction on Transactions on Circuits and Systems for Video Technology (TCSVT), 2015. (SCI收录)

[2] **Xie L**, Tian Q, Zhou W, Zhang B. Heterogeneous Graph Propagation for Large-Scale Web Image Search. *to appear*, in IEEE Transaction on Image Processing (TIP), 2015. (SCI收录)

[3] **Xie L**, Wang J, Zhang B, Tian Q. Fine-Grained Image Search. *to appear*, IEEE Transaction on Multimedia (TMM), volume 17, number 5, pages 636–647, 2015. (SCI收录)

[4] **Xie L**, Tian Q, Wang M, Zhang B. Spatial Pooling of Heterogeneous Features for Image Classification. IEEE Transaction on Image Processing (TIP), volume 23, number 5, pages 1994–2008, 2014. (SCI收录, DOI: 10.1109/TIP.2014.2310117)

[5] **Xie L**, Tian Q, Zhou W, Zhang B. Fast and Accurate Near-Duplicate Web Image Search with Affinity Propagation on the ImageWeb. Computer Vision and Image Understanding (CVIU), volume 124, pages 31–41, 2014. (SCI收录, DOI: 10.1016/J.CVIU.2013.12.011)

[6] **Xie L**, Hong R, Zhang B, Tian Q. Image Classification and Retrieval are ONE. *to appear*, ACM International Conference on Multimedia Retrieval (ICMR), Shanghai, China, 2015.6. (EI收录)

[7] **Xie L**, Tian Q, Wang, J, Zhang B. Image Classification with Max-SIFT Descriptors. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, Australia, 2015.4. (EI收录)

[8] **Xie L**, Tian Q, Zhang B. Generalized Regular Spatial Pooling for Image Classification. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, Australia, 2015.4. (EI收录)

[9] **Xie L**, Tian Q, Zhang B. Max-SIFT: Flipping Invariant Descriptors for Web Logo Search. IEEE International Conference on Image Processing (ICIP), pages 5716–5720, Paris, France, 2014.10. (EI收录)

[10] **Xie L**, Wang J, Guo B, Zhang B, Tian Q. Orientational Pyramid Matching for Recognizing Indoor Scenes. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), pages 3734–3741, Columbus, USA, 2014.6. (EI收录)

[11] **Xie L**, Tian Q, Hong R, Yan S, Zhang B. Hierarchical Part Matching for Fine-Grained Visual Categorization. IEEE International Conference on Computer Vision (ICCV), pages 1641–1648, Sydney, Australia, 2013.12. (EI收录)

[12] **Xie L**, Tian Q, Zhang B. Feature Normalization for Part-Based Image Classification. IEEE International Conference on Image Processing (ICIP), pages 2607–2611, Melbourne, Australia, 2013.9. (EI收录)

[13] **Xie L**, Tian Q, Zhang B. Spatial Pooling for Heterogeneous Features for Image Applications. ACM International Conference on Multimedia (ACM-MM), pages 539–548, Nara, Japan, 2012. (EI收录)

在学期间参加课题的研究成果

本人于2008年12月参加计算机代数系统maTHμ课题小组，担任前端开发负责人，并于2009年4月完成maTHμ的第一个前端版本。学生立项项目“计算机代数系统maTHμ”先后获得清华大学第二十七届“挑战杯”学生课外学术科技作品竞

赛特等奖、第五届“挑战杯”首都大学生课外学术科技作品竞赛特等奖以及第十一届“挑战杯”（航空航天）全国大学生课外学术科技作品竞赛特等奖。