

北京郵電大學

本科 毕业 设计（论文）



题目：基于卷积神经网络的图像检索系统的设计与实现

姓 名 张冀韬
学 院 信息与通信工程学院
专 业 信息工程
班 级 2014211125
学 号 2014210669
班内序号 16
指导教师 张彬

2018 年 6 月

北 京 邮 电 大 学

本科毕业设计（论文）诚信声明

本人声明所呈交的毕业设计（论文），题目《基于卷积神经网络的图像检索系统的设计与实现》是本人在指导教师的指导下，独立进行研究工作所取得的成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他教育机构的学位或证书而使用过的材料。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名：_____ 日期：_____

基于卷积神经网络的图像检索系统的设计与实现

摘 要

随着移动互联网的兴起，以图片和视频为主要载体的各类多媒体数据成爆炸式的增长。如何在浩瀚的图片数据库中，快速，方便的查找出用户需要并且感兴趣的图片成为了现如今各大研究机构和科技巨头的竞争方向。同时由于数据的指数增长，伴随着计算能力的不断提高，卷积神经网络技术成为了图像检索领域的新宠。不同于传统的特征提取和图像检索算法，神经网络自身就带有提取基于内容的检索特征的特点，网络的从浅入深能够帮助我们从高维到低维，低语义到高语义的提取适合的特征，配合相应的检索算法，可以帮助我们进行高效的检索。

本文设计和实现了基于卷积神经网络的图像检索系统，主要进行以下工作：

- 1、探究卷积神经网络的实现原理和相关优化训练算法。
- 2、探究卷积神经网络的 Fine-tune 技术在多维度特征提取技术中的应用。
- 3、实现基于神经网络的图像检索系统。使该系统支持库内检索和上传检索两种方式。库内检索将根据库中任意图片的特征利用检索算法和设定阈值进行检索，上传检索将对新图片进行特征提取从而进行检索。同时在系统中引入了二值哈希特征，在确保准确率不大幅度降低的同时将检索分为粗筛选和细筛选，大大提高了检索效率。
- 4、展示系统的实现，分析图像检索系统的准确率，平均精度均值和时间降比等指标。

关键词 图像检索 卷积神经网络 哈希化 特征提取

DESIGN AND IMPLEMENTATION OF AN IMAGE RETRIEVAL SYSTEM BASED ON CONVOLUTIONAL NEURAL NETWORK

ABSTRACT

With the rise of mobile Internet, all kinds of multimedia data, which are mainly supported by pictures and videos, have become explosive growth. How to quickly and conveniently find out the needs and interests of the users in the vast image database has become the competitive direction of the major research institutions and the science and technology giants. At the same time, due to exponential growth of data, convolution neural network technology has become the new favorite of image retrieval with the continuous improvement of computing power. Different from the traditional feature extraction and image retrieval algorithms, the neural network itself has the characteristics of extracting the content based retrieval features. The network from shallow to depth can help us extract the suitable features from high dimension to low dimension, low semantic to high semantic, and with corresponding retrieval algorithms, it can help us to carry out high efficiency in retrieval process. In this paper, an image retrieval system based on convolution neural network is designed and implemented.

- 1.Explore the realization principle and related optimization training algorithm of convolution neural network.
- 2.Explore the application of convolution neural network Fine-tune technology in multi-dimensional feature extraction technology.
- 3.Implement the image retrieval system based on convolutional neural network. The system supports two ways of searching, uploading or searching in the library. In library retrieval, the relevant images will be retrieved based on the retrieval algorithm and setting threshold according to the features of any picture in the library. In uploading retrieval, the features of the new image will be extracted and then the relevant images will be retrieved the same as library retrieval. At the same time, binary hash features are introduced in the system, and the retrieval is divided into coarse searching and fine searching. At the same time the accuracy is not reduced greatly and the retrieval efficiency is greatly improved.
- 4.Display the CIBR system and analysis of image retrieval system based on accuracy, mean average precision and drop ratio of time or other indicators.

KEY WORDS image retrieval convolution neural network hash method feature extraction

目 录

第一章 绪论	1
1.1 选题的背景和意义	1
1.2 国内外研究现状	2
1.3 论文的主要工作	2
1.4 论文的组织结构	3
1.5 本章小结	3
第二章 相关技术简介	4
2.1 检索系统数据集	4
2.2 检索系统评估标准	6
2.2.1 准确率.....	7
2.2.2 精确率与召回率.....	7
2.2.3 F1 值	8
2.2.4 AP 与 MAP 指标	8
2.3 现代特征提取方法	8
2.3.1 HOG 特征	8
2.3.2 LBP 特征.....	9
2.3.3 SIFT 特征	9
2.4 相似性度量公式	9
2.4.1 卡方距离.....	9
2.4.2 汉明距离.....	9
2.4.3 欧式距离.....	10
2.4.4 曼哈顿距离.....	10
2.4.5 切比雪夫距离.....	10
2.5 近代检索算法	10
2.5.1 词袋模型.....	10
2.5.2 VLAD	11
2.6 本章小结	11
第三章 特征提取与检索的相关算法	12
3.1 卷积神经网络基础	12
3.1.1 卷积层简介.....	12
3.1.2 防止过拟合.....	12

3.1.3 高效训练.....	13
3.2 卷积神经网络结构	14
3.3 微调	15
3.3.1 为什么要微调.....	15
3.3.2 什么情况下使用微调.....	15
3.3.3 微调的注意事项.....	15
3.3.4 微调与迁移学习的区别.....	15
3.4 特征提取和检索	16
3.4.1 数据预处理.....	16
3.4.2 特征提取与检索.....	16
3.4.3 Fine-tune 模型训练.....	18
3.5 本章小结	19
第四章 检索系统的功能设计.....	20
4.1 系统的技术框架	20
4.2 系统的工作流程	21
4.2.1 库内检索.....	21
4.2.2 上传检索.....	21
4.3 关键模块分析	21
4.3.1 客户端.....	21
4.3.2 服务端.....	22
4.3.3 检索系统.....	22
4.3.4 数据库.....	23
4.4 本章小结	24
第五章 检索系统的实现.....	25
5.1 系统的主界面	25
5.2 库内检索	26
5.3 上传检索	27
5.4 系统结果分析	28
5.5 本章小结	31
第六章 总结和展望.....	32
6.1 总结	32
6.2 展望	32
参考文献.....	33
致 谢.....	34

第一章 绪论

1.1 选题的背景和意义

随着智能手机和平板电脑等智能终端的大规模普及，移动互联网开始迅猛发展。伴随与此，尤其是以图片和视频等多媒体数据也在呈爆炸指数级的增长。相较于普通的文本搜索，图像检索变得越来越重要。在如此浩瀚庞大的图像数据库中如何快速，方便的检索出用户感兴趣的图片已经成为了各大科技巨头和研究机构开始着手解决的主要问题。

最早的图像检索方式主要是基于文本的图像检索方式，于上个世纪 70 年代提出，这种方式简单直接，以人工标注的方式将图片人为的根据种类，背景，内容等一系列方面打上标签，赋予其意义。用户在交互界面中输入感兴趣的话题作为关键字，系统则以关键词的形式与图像数据库中的每一张图片的标签进行相似性对比，同时对相似性度量结果进行排序并返回相似度较高的图片作为结果。顾名思义，基于文本的图像检索方式更多的是借助于文本处理的技术对图像进行分类。而在现如今，这种方式的局限性已经逐渐暴露出来。这种方式对于小规模图片集往往是可行的而且准确率很高。但是随着数据量的暴增，线性的人工标注速度已经不可能跟上数据指数级增长的速度，并且受制于人的文化认知和思想差异，标注的差异性过大也会同时降低查找的准确率。同时，在浩瀚的图像数据库中，也有一定比率的图片很难给予少数的关键词进行描述。根据上述的几个考量因素，这种费时费力并且不能保证效率和准确率的方式在如今的大环境下必然会被淘汰。

于是，在上个世纪 90 年代，基于内容的图像检索就应运而生。这种检索方式则是尽可能的希望使用算法利用计算机自动的提取出“标签”，这种标签并不是人可以阅读理解的，而是计算机根据算法对于图片的高层语义抽象。我们将这种高层语义抽象称为图片的特征。同时自动提取的特征过程也避免了人工的主观臆断。在进行检索时，对于待检索的图片我们以相同的特征提取方式进行提取，以某种相似性算法进行对比，设定阈值返回相似性最高的一系列图片。这种方式有很明显的劣势就是对于特征的自动提取，大大利用了计算机的计算优势。但是缺陷也在于，人类所浓缩的极高层语义和计算机所提取的语义还是有很大的差别，至少在现如今还是难以消除。

尽管如此，图像检索还是应用到了我们生活中的方方面面。在网上商城中，阿里巴巴提供了上传商品图片检索商品的功能，在我们不知道商品名称的情况下，大大帮助了用户快速找到类似商品或直接进行商品的精确定位。在公安系统中，对于已经有犯罪前科的犯罪分子，公安人员可以将其肖像，指纹，手印等以图片形式存入数据库，以防下次该嫌犯再犯，就可以快速的在数据库中进行比对确认嫌疑人。在版权保护中，对于新

商标注册的判重检索，可以避免版权纠纷等事件。在浩瀚的商标图像库中找到类似的商标图进行判重分析，图像检索的效率和准确率就至关重要。在医疗诊断领域，医生可以将患者的诊断图片在确诊库中进行检索，根据相似图像的诊断结果，综合考虑对新患者进行诊断分析。

图像检索已经在我们生活的方方面面提供了便利，相信在未来，伴随着计算能力的进一步提高和新的特征提取和检索算法的提出，图像检索会变得越来越智能化。

1.2 国内外研究现状

国外的学者在图像检索领域，尤其是对于相同物体检索已经提出了很多算法。主要是分为早期的低层语义和现今的高层语义。

在图像检索领域的初期，特征提取器主要是提取图像的低层特征比如基于颜色提取特征，基于纹理提取特征，基于形状提取特征以及基于上述三类进行多维度的特征提取。基于颜色提取的特征主要可以降低因为旋转，平移，尺度变换所带来的干扰，因为像素具有稳定性，从而该特征提取算法也表现出了相当强的鲁棒性。在基于纹理特征的特征提取上，主要是利用了相邻的像素点的对比反映出的有规律的变化，其具有周期性的同时也具有一定的统计特性。基于形状的特征提取主要是对于图像的轮廓边界进行描述作为特征。

现如今，在基于内容的图像检索领域，研究人员越来越关注如何提取出图像内部的高层语义。而不是仅仅将像素孤立，提取基于颜色的特征，或者基于简单的相邻像素点规律提取纹理特征，亦或者将图像分割成封闭的区域提取基于形状的特征。图像本质上是一个实体的对象，蕴含着丰富的感情色彩和高层语义。现在比较流行的图像特征提取算法譬如 SIFT^[1], SURF^[2], Fisher^[3]向量等，都是基于提取抗干扰较好的不变性局部特征，大体上讲，可以得到很高的检索精度，但往往提取的特征维度太高，有时甚至高达几十万维，在海量图像检索的情况下就需要更加高效的检索算法提高效率。

此领域的新宠就是卷积神经网络，这种网状的模拟人脑的自动提取特征的算法能够帮助我们显式的提取从高维到低维的特征，对于大部分图片最后得到 4096 维特征，再结合 PCA 降维和配套相应的检索算法，使之适应大规模海量数据集就成为了可能。

1.3 论文的主要工作

论文主要介绍了一个基于卷积神经网络的图像检索系统的设计和实现。详细的工作可以描述如下：

一、明确选题的主要内容：设计一个基于卷积神经网络的图像检索系统。选题既要侧重于特征提取和高效检索算法的选取，也要侧重于数据的流通过程，即系统前后端和数据库的合理搭建与交互方式。

二、分析检索系统的特征提取和检索引擎的功能及其实现方式，选取合适的技术栈进行搭建。

三、提出系统的详细设计方案。包括系统的整体架构，数据的运行过程。深入分析系统的各功能模块，阐述模块的实现过程。

四、展示系统的实现。演示系统，从精确率，时间降比和平均精度均值等指标分别分析系统。

五、总结分析系统仍存在的缺陷和可能的改进方案，继续完善。

1.4 论文的组织结构

第一章是绪论。该部分主要介绍选题的背景和意义，国内外研究现状，论文的主要工作和论文的组织结构。

第二章是相关技术简介。该部分介绍数据集概况，系统评估的相关指标，特征提取的相关算法和相似性度量的常用距离与近代的检索算法。

第三章是特征提取与检索的相关算法。该部分由浅入深，首先简述卷积神经网络的大体运行方式和训练技巧，之后着重介绍如何利用 fine-tune 进行特征提取并且介绍系统采用的二值哈希算法和相似性度量算法。

第四章是检索系统的功能设计。该部分介绍系统的整体框架，描述系统的前端，服务端，数据库和检索系统之间如何进行交互，对关键的功能模块进行详细分析。

第五章是检索系统的实现。该部分演示系统的实际效果并分别对功能点进行举例，同时对于系统进行指标分析。

第六章总结。该部分为论文做总结并提出可能的改进方向。

1.5 本章小结

本章主要介绍了本课题的选题背景和意义，图像检索在国内外的研究进展，论文的主要工作和组织结构。

第二章 相关技术简介

2.1 检索系统数据集

为了更好的保证数据的多样性，本检索系统采用的数据集是 Caltech101 图片数据集，共 101 类 8677 张图片。每一类的图片有约 40-800 张不等。大多数类有 50 张图片，并且分布类型广泛，涵盖了风景，动物，植物，人脸，机械等多种类别。每一张图像的像素大约为 300×200 像素。

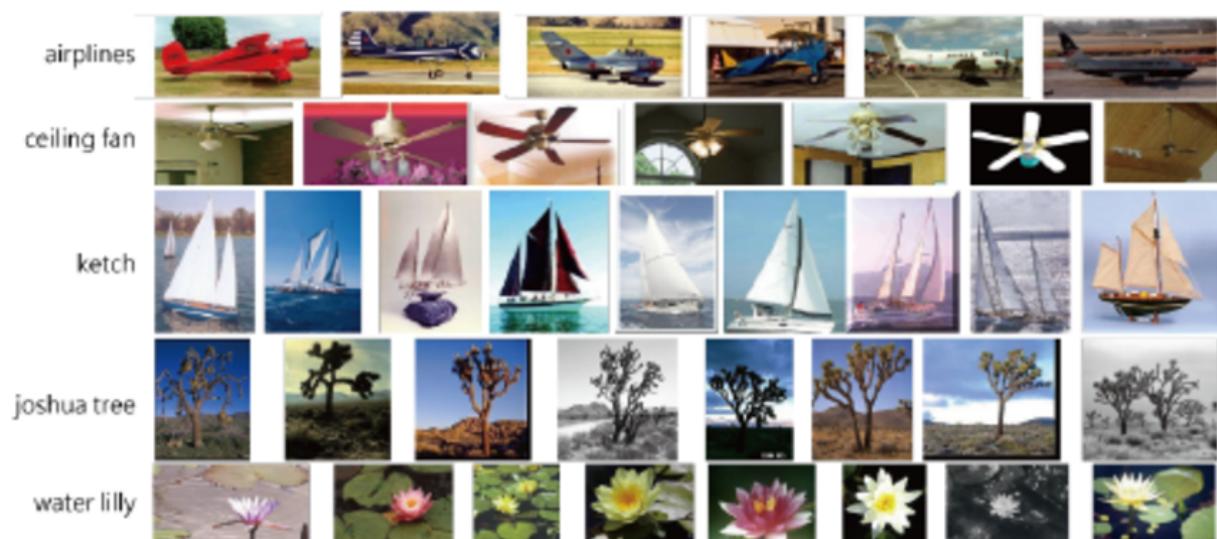


图 2-1 图片举例

表 2-1 数据集类名

ID	名称	ID	名称	ID	名称	ID	名称
1	accordion	27	cup	53	kangaroo	79	saxophone
2	airplanes	28	dalmatian	54	ketch	80	schooner
3	anchor	29	dollar_bill	55	lamp	81	scissors
4	ant	30	dolphin	56	laptop	82	scorpion
5	barrel	31	dragonfly	57	Leopards	83	sea_horse
6	bass	32	electric_guitar	58	llama	84	snoopy
7	beaver	33	elephant	59	lobster	85	soccer_ball
8	binocular	34	emu	60	lotus	86	stapler
9	bonsai	35	euphonium	61	mandolin	87	starfish
10	brain	36	ewer	62	mayfly	88	stegosaurus
11	brontosaurus	37	Faces	63	menorah	89	stop_sign
12	buddha	38	Faces_easy	64	metronome	90	strawberry
13	butterfly	39	ferry	65	minaret	91	sunflower
14	camera	40	flamingo	66	Motorbikes	92	tick
15	cannon	41	flamingo_head	67	nautilus	93	trilobite
16	car_side	42	garfield	68	octopus	94	umbrella
17	ceiling_fan	43	gerenuk	69	okapi	95	watch
18	cellphone	44	gramophone	70	pagoda	96	water_lilly
19	chair	45	grand_piano	71	panda	97	wheelchair
20	chandelier	46	hawksbill	72	pigeon	98	wild_cat
21	cougar_body	47	headphone	73	pizza	99	windsor_chair
22	cougar_face	48	hedgehog	74	platypus	100	wrench
23	crab	49	helicopter	75	pyramid	101	yin_yang
24	crayfish	50	ibis	76	revolver		
25	crocodile	51	inline_skate	77	rhino		
26	crocodile_head	52	joshua_tree	78	rooster		

表 2-2 各类数目

ID	数 量	ID	数 量								
1	55	18	59	35	64	52	64	69	39	86	45
2	800	19	62	36	85	53	86	70	47	87	86
3	42	20	107	37	435	54	114	71	38	88	59
4	42	21	47	38	435	55	61	72	45	89	64
5	47	22	69	39	67	56	81	73	53	90	35
6	54	23	73	40	67	57	200	74	34	91	85
7	46	24	70	41	45	58	78	75	57	92	49
8	33	25	50	42	34	59	41	76	82	93	86
9	128	26	51	43	34	60	66	77	59	94	75
10	98	27	57	44	51	61	43	78	49	95	239
11	43	28	67	45	99	62	40	79	40	96	37
12	85	29	52	46	100	63	87	80	63	97	59
13	91	30	65	47	42	64	32	81	39	98	34
14	50	31	68	48	54	65	76	82	84	99	56
15	43	32	75	49	88	66	798	83	57	100	39
16	123	33	64	50	80	67	55	84	35	101	60
17	47	34	53	51	31	68	35	85	64		

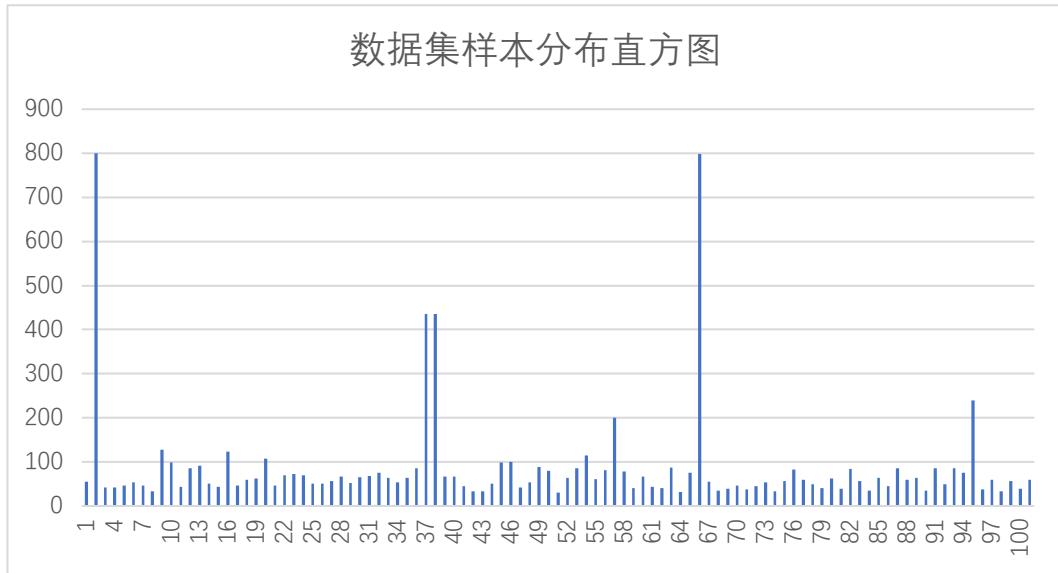


图 2-2 数据集样本分布直方图

2.2 检索系统评估标准

检索系统在完成检索后是需要进行评估反馈的，有助于下一次的迭代。但由于系统任务需求的不同，对于检索结果的“相关性”的定义就不同，所以需要的评价指标也不

同。对于检索系统的评价指标往往是多元的，即为了进行评估时更加有侧重性。以下将介绍几种常用的评价指标。

2.2.1 准确率

准确率(accuracy)的定义是，对于给定的测试数据集，预先训练好的分类器分类正确的样本数与总样本数的比值。即对于给定数据集 $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ ，其中 y_i 是样本 x_i 的真实类别， I 作为真值函数，在预测函数 f 下，将准确率定义为

$$\text{Acc}(f; D) = \frac{1}{m} \sum_{i=1}^m I(f(x_i) = y_i) \quad \text{式(2-1)}$$

准确率定义简单但有局限性。举一个简单的例子：假如要在 Caltech101 训练集中寻找与飞机类型的第一张图片最相似的 100 张图片，同时库中共有 800 张飞机类图片，检索结果为 100 张中有 50 张为飞机，则准确率高达 $\frac{8677-50}{8677} = 99.4\%$ ，貌似是一个非常好的结果，但由于总样本数过高，所以并不能将检索系统的精度完全的度量出来。

2.2.2 精确率与召回率

为了弥补准确率的不足，精确率(precision)与召回率(recall)对于评价检索系统的性能就变得至关重要。在定义这两个指标之前，需要对每一次检索的所有样本进行分类，共有四类，分别是 True Positives(相关的样本被检索出来的个数)，False Positives(无关样本被检索出来的个数)，False Negatives(相关样本未被检索出来的个数)，True Negatives(无关样本未被检索出来的个数)。

表 2-3 精确率与准确率定义

	相关, 正类	无关, 负类
被检索到	TP	FP
未被检索到	FN	TN

假设精确率为 P，则精确率计算公式：

$$P = \frac{TP}{TP + FP} \quad \text{式(2-2)}$$

即在被检索出来的结果样本中计算相关结果的占比，上例中该结果即 $50/100=50\%$

假设召回率为 R，召回率计算公式：

$$R = \frac{TP}{TP + FN} \quad \text{式(2-3)}$$

即在总相关样本中计算被检索出的样本数的占比，上例中该结果为 $50/800=6.25\%$

但是这两个指标在一些极端情况下往往是矛盾的，因为当精确率高而检索总数目很少时，这时召回率将会变得非常低，而当召回率很高且检索总数目很多时，精确率将会变得非常低。

2.2.3 F1 值

为了解决上述的问题，引入了新的指标称为 F1-Measure，对精确率和召回率做调和平均。 β 作为调和参数，P 为精确率，R 为召回率。

通用形式：

$$F_{\beta} = (1 + \beta^2) \cdot \frac{P \cdot R}{(\beta^2 \cdot P) + R} \quad \text{式(2-4)}$$

常用形式将 β 取 1 得到下式

$$F_{\beta} = \frac{2}{\frac{1}{R} + \frac{1}{P}} = 2 \cdot \frac{P \cdot R}{P + R} \quad \text{式(2-5)}$$

在对于精确率和召回率都有严格要求的系统中，这个指标将帮助工程人员进行数据分析从而提高系统的整体检索效率。

2.2.4 AP 与 MAP 指标

平均精度 (Average Precision) 和平均精度均值 (Mean Average Precision) 是两个在文档检索和推荐系统中广泛运用的指标，在图像检索中同样可以作为参考。这种评价方式主要结合了 ranking 进行了分析。

$|R|$ 代表本次检索结果中相关结果的总数， $Prec(i)$ 代表 top i 位的精确率， $relevance(i)$ 表示在 i 位的相关情况，若相关取 1，否则取 0。则这一次检索的平均精度为

$$AP = \frac{1}{|R|} \sum_{i=1}^n Prec(i) \cdot relevance(i) \quad \text{式(2-6)}$$

假设共检索了 Q 次，第 i 次的检索记为 Q_i ，平均精度为 $AP(Q_i)$ ，则平均精度均值为每一次的平均精度加和求平均。即

$$MAP = \frac{1}{|Q|} \sum_{Q_i \in Q} AP(Q_i) \quad \text{式(2-7)}$$

总的来说，平均精度衡量的是检索系统在某个类别或者某次检索的好坏情况，平均精度均值是衡量对于所有类别的好坏情况，即整个系统的检索效率。

2.3 现代特征提取方法

2.3.1 HOG 特征

方向梯度直方图 (Histogram of Oriented Gradient, HOG^[4]) 特征是人为定义的一种特征描述方式，该方法利用图像的梯度方向构成直方图来形成特征，同时与支持向量机分类器相配合，在图像识别领域取得了不错的成果。其主要思想是将图像分割成连通的小单元，称其为细胞单元，之后将各像素点的梯度提取出来构成直方图作为特征描述器。

由于是采用的局部特征提取，所以对于图像几何和光学的形变都有很好的抗干扰能力，鲁棒性较强。

2.3.2 LBP 特征

局部二值模式 (Local Binary Pattern, LBP^[5]) 特征是基于局部的纹理特征的算子。该特征基于每一个像素点与周围邻域上的像素点的灰度关系得到该像素点的特征，所以具有很好的旋转不变性和灰度不变性。

2.3.3 SIFT 特征

尺度不变特征变换 (Scale-Invariant feature transform, SIFT) 同样是基于局部的特征描述。其特殊之处就在于特征的提取可以排除图片尺度的缩放，旋转，亮度等干扰。实现方式较为复杂，首先需要对图像构建尺度空间，检测极值点，获得尺度不变性。之后对特征点进行过滤并为特征点分配方向值生成特征，在这一步生成时为了确保旋转不变性和光照影响分别会以关键点的方向旋转坐标轴并且将特征向量的长度归一化。最终通过欧氏距离的大小判断相似性。类 SIFT 的特征提取方式有很多，比如 SURF，ORB^[6]，VLAD^[7]等，都是采用了由局部到全局的特征表达方式。但特征的维度往往是十分高的，所以高效的索引机制就变得十分重要。

2.4 相似性度量公式

对于两个向量特征的相似性度量方式有很多，选取合适的度量公式也是必要的。以下是一些常用的相似性度量方式。

2.4.1 卡方距离

卡方距离是一种计算两个列表差异性的一种统计量。在图像领域，也得到了应用。假设两个离散向量为 x_u, x_v ，其均为 n 维，对应位值为 $x_u(n), x_v(n)$ ，计算公式如下

$$d(x_u, x_v) = \sum_{i=1}^n \frac{[x_u(i) - x_v(i)]^2}{x_u(i) + x_v(i)} \quad \text{式(2-8)}$$

在系统中进行细筛选时，需要对两个 4096 维的特征向量使用卡方距离计算相似性。

2.4.2 汉明距离

汉明距离即为两个二进制字符串按位比较的不同字符的个数。假设有向量 i, j 长度均为 n，对应位值分别为 $y_{i,k}, y_{j,k}$ 。则距离为

$$d^{HAD}(i, j) = \sum_{k=0}^{n-1} y_{i,k} \neq y_{j,k} \quad \text{式(2-9)}$$

在系统中进行粗筛选时，需要对待检索图片的哈希特征和数据库中所有的去重后的所有哈希特征进行汉明距离的度量，对在阈值内的哈希特征所对应的所有图片再进行细检索。

2.4.3 欧式距离

欧式距离最易于理解，对于两个给定的 n 维向量，对应位的差值平方求和即为欧式距离。假设存在向量 $P = (p_1, p_2, \dots, p_n)$ 与 $Q = (q_1, q_2, \dots, q_n)$ ，欧式距离为

$$d(p, q) = d(q, p) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad \text{式(2-10)}$$

2.4.4 曼哈顿距离

曼哈顿距离是根据现实生活中的街区路线所命名的。人在跨越街区时由于建筑物的妨碍不能以直线达到目的地，只能按照坐标系的各个方向移动。对于给定的 n 维向量，对应位做差的绝对值求和即为曼哈顿距离。假设存在向量 $P = (p_1, p_2, \dots, p_n)$ 与 $Q = (q_1, q_2, \dots, q_n)$ ，曼哈顿距离为

$$d(p, q) = d(q, p) = \sum_{i=1}^n |p_i - q_i| \quad \text{式(2-11)}$$

2.4.5 切比雪夫距离

切比雪夫距离是两个给定 n 维向量对应位做差求绝对值的最大值。

$$d(p, q) = \max_i(|p_i - q_i|) \quad \text{式(2-12)}$$

2.5 近代检索算法

2.5.1 词袋模型

词袋模型 (BOW^[8]) 主要应用在自然语言处理，信息检索，计算机视觉等多个方面。主要是为了解决文档的分类问题。对于文本，先将所有的文本中独立的单词取出作为词典并设定标记。之后将待检索的文本在词典中做映射形成特征向量进行相似性度量。该方法对于图像检索同样可以适用。通常首先对所有图片提取不同数目的 128 维 SIFT 特征，之后使用 K-means 聚类算法对所有的特征进行聚类，分成不同的簇，使簇间的相似性尽量低，簇内的相似性尽量高，同时我们也可以将每一个簇看做词典中的一个词，将所有簇的集合看作码本。最后计算每一幅图像的每一个 SIFT 特征到码本中的每一个簇的距离，就可以得到每一张图片的词频矢量。这种算法的巨大优势就是降维，否则对于成百上千的 SIFT 矢量进行相似性对比往往是十分耗时的。但是该方法也有一定缺点，

即如果对于码本的数目有限制，则码本的特征表达会越来越粗略，导致特征的许多信息被丢失。

2.5.2 VLAD

VLAD 与词袋模型的大致流程非常相似。首先都要对所有的图片提取 SIFT 特征向量，进而使用 K-means 进行聚类得到码本，但与词袋模型不同之处在于 VLAD 会统计落入最近单词与该单词的累计残差。简单地说 VLAD 并没有直接用码本中的一个词去代替该特征，而是得到了码本中所有词与该特征的相似性距离作为新的特征向量，所以并没有丢失大量信息。

2.6 本章小结

本章首先主要介绍了本课题选用的数据集的原因及其具体分布情况。然后从简单到复杂介绍了图像检索领域的评估指标，包括最简单的准确率，精确率和召回率，至比较复杂的平均精度和平均精度均值。之后对现今比较流行的特征提取方案和检索方案进行了简单说明，也介绍了几种相似性度量距离的方式。

第三章 特征提取与检索的相关算法

3.1 卷积神经网络基础

普通的神经网络不适用于大规模高清晰度的图片识别。原因有二，一是计算能力还没有达到能够训练大量高清晰度的照片，如果单纯的将图片的像素点的数量作为维度，一张人眼都难以分辨的低清晰度的图片在很深的神经网络下，单单一层就会可能需要训练百万甚至千万个权重。这是不现实的。同时也是在现如今系统需要快速迭代的需求背景下是不能接受的。所以对于清晰度高的图片肯定就无法进行高效的识别。二是以传统通信的角度，图片是二维的信号，凡是二维的信号一般就具有相关性，单纯的将二维图片压缩成一维在普通的神经网络中训练也会丧失掉这种相关性，在某种程度上也是降低了检索效率的上限。卷积神经网络的出现就是要将训练神经网络的方式结合传统特征提取的算法进行更加高效和针对性的训练。

3.1.1 卷积层简介

相比于普通的神经网络只有隐藏层，卷积神经网络有三种类型的隐藏层，分别是卷积层，池化层，全连接层。核心就是不断地在进行特征提取的同时又进行降维。卷积层顾名思义与传统通信中的图像检测中的算子相同，从二维来看是可以在两个方向移动的卷积区域，由于图片往往有 RGB 三个通道，所以卷积核往往也是三维的，最后一个维度是 3 与 RGB 三个通道相对应。池化层的作用往往是在多个卷积层后进行再一次的降维，常用的有最大池化层和平均池化层。顾名思义就是取卷积区域的最大值或者平均值作为下一层的对应位置的输出。全连接层类似于普通神经网络的隐藏层，往往在进行不同维度的特征提取时，进行指定维度的降维用于进行特征的输出。

3.1.2 防止过拟合

过拟合顾名思义就是对于训练数据集太过于相信，往往使得训练集识别率太高而在测试集上表现不好的一种状态。简而言之，就是卷积神经网络过于“敏感”和“聪明”，使得神经网络的参数貌似是对于特定的训练数据集的“过度附和”。在对于卷积神经网络的训练中，采取的主要方法有两个。第一个是正则化 (Regularization)，第二个是随机失活 (Dropout^[9])。其根本思想就是降低神经网络的敏感度，也就是在可控范围内对卷积神经网络进行“破坏”。

(1) 正则化 (Regularization)

$$\frac{\lambda}{2m} \sum_{l=1}^L \|\omega_l\|^2 \quad \text{式(3-1)}$$

$$J(w) = \frac{1}{m} \sum_{i=1}^m l(y^{(i)}, \hat{y}^{(i)}) + \frac{\lambda}{2m} \sum_{l=1}^L \|w_l\|^2 \quad \text{式(3-2)}$$

λ 为正则化的超参数， w_l 为第 l 层权重矩阵， $y^{(i)}$ 为实际值， $\hat{y}^{(i)}$ 为预测值， m 为训练样本个数。

常规的损失函数为式 3-2 等式右边的左半部分，L2 正则化会在右边加入一个惩罚项，也就是式 3-1，来进行正则化。求导后所求的梯度会将 w_l 之前的系数由 1 变为 $1 - \frac{\alpha\lambda}{m}$ ，这也是为什么 L2 正则化又被称为权值衰减（Weight Decay），原因是最后反向传播时权重矩阵 W 里面的元素会减小，乘以一个接近于 1 但小于 1 的常数，并且这个降低的幅度可以通过控制 λ 进行更改。

L2 正则化能够防止过拟合的原因就在于此。如果将超参数 λ 选的足够大，则权重矩阵中的许多参数会变得极小，直观地说就是神经网络中的许多神经元会“不起作用”，此方法既没有降低神经网络的广度，也没有降低深度。

(2) 随机失活 (Dropout)

反向随机失活。即将某一层的参数矩阵随机的以一定的概率 (keep-prob) 作为比例，失活其中的因子，简单的说就是将其中的一些参数置 0。但是为了不改变期望均值，还要将存活下来的神经元除以之前的概率 (keep-prob)。这个方法会降低卷积神经网络的广度，但不会降低深度。

3.1.3 高效训练

(1) 随机梯度下降法 (Stochastic Gradient Descent)

在大训练集下，使用普通的梯度下降法往往对于损失函数的收敛速度较慢，原因是在每一次反向传播中都要遍历所有点的梯度进行下降，但往往能到达局部最优点。但对于需要效率的我们，使用随机梯度下降在每次反向传播中只随机选取一个点进行梯度下降，尽管很难到达局部最优点，但一定能快速的接近局部最优点。这也是在大样本训练中考虑到计算成本的折中方法。

(2) 数据标准化 (Batch Normalization^[10])

总的来说，就是尽量保证不同维度输入特征的尺度相同。使得损失函数能够在不同的方向上都快速收敛，加快学习过程。但批处理的归一化和普通神经网络在输入层进行归一化还有所不同，区别是在隐藏层也会进行对激活前的 z 值的归一化，而且不一定是输入层标准的均值为 0 方差为 1 的归一化，而是利用两个超参数去学习得到的该层的均值和方差，并且适配于其指定的激活函数。

(3) 批处理 (Mini-batch Training)

往往在训练数据集过大的情况下，一次训练就要遍历所有的数据，假如有 5000000 个数据，那么梯度下降一次耗费的时间就过于长，训练的效率自然就降低了。如果我们

将 1000 个数据看为一个小的训练集，则大的数据集将被视为 5000 个小数据集，这样在遍历完一次数据集后，梯度进行了 5000 次更新。大大加快了训练的速度。但如果观察 cost-epoch 图，

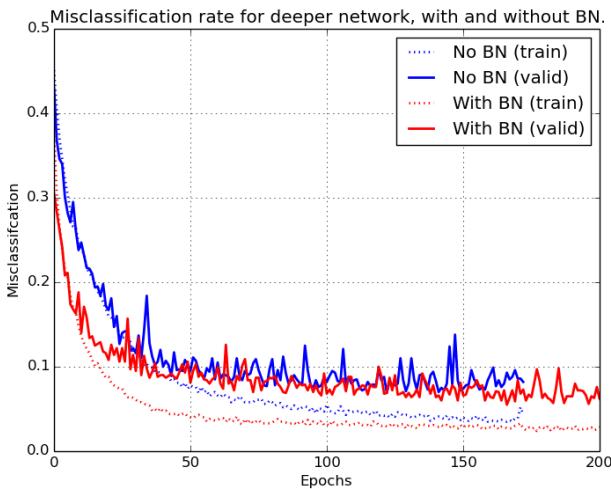


图 3-1 cost-epoch

则会发现损失函数虽然大趋势下会下降，但明显的有噪声。但也能够保证损失函数收敛到接近于最低点。在选取 mini-batch 的大小时也要注意，如果太小会变成随机梯度下降，使得损失函数很难收敛到全局最小值甚至收敛不到，太大则失去了批处理的意义。一般选取 64, 128, 256, 512 作为 mini-batch 的大小。还有一定要确保每一次的小数据集的大小在神经网络的不同层占用的内存要小于 CPU 或者 GPU 的内存大小，否则程序会在中途崩溃。

(4) 数据增强 (Data Augmentation)

数据增强是一种简单的增加数据规模的方式。可以通过对于原图的操作，使你足不出户也可以成倍的扩大训练集。总的来说有以下几种方式：镜像，裁切，颜色变换。在 Keras 中也提供了相关的 API，只需要指定几个参数就可以将原训练集进行增强。

3.2 卷积神经网络结构

本课题中选定的是 VGG19^[11]卷积神经网络。一共有 19 层，通常计算时不包括池化层。VGG 卷积层使用 $3 * 3$, stride 为 1, SAME 的卷积核，池化层选择 $2 * 2$, stride 为 2 的卷积核。每做一次卷积，原始图像的矩阵宽高不变，而通道 (channel) 数与卷积核的个数相同，19 层分别为 $2 * 64$, $2 * 128$, $4 * 256$, $4 * 512$, $4 * 512$ 。共 16 层，如果加上最后的两层全连接层 (fully-connected layer) 和分类层 (softmax layer) 就是 19 层。上述的每一次间隔都会做一次池化，由于池化选择的卷积核为 $2 * 2$ ，所以经过池化后的原三维图像矩阵宽高会减半而通道数不变。

选择该模型的原因有二：

一是几个小滤波器卷积层的组合比一个大滤波器卷积层好。小的卷积核连用和具有相同感受野的大卷积核相比，小卷积核由于深度更深，就具有多层的非线性函数，更加具有判决性。

二是在之后的训练中会主要以 Fine-tune 为主，VGG19 在我们选取的深度学习框架 Keras 中有已经训练好的权重文件，能够方便我们更快的进行模型修改，从而提取特征进行试验。

3.3 微调

3.3.1 为什么要微调

卷积神经网络的核心就是浅层卷积层提取基础特征，类似于颜色，边缘，轮廓，纹理等基础特征。深层的卷积层提取较为抽象的特征，比如图像的背景，人脸的整个脸型等。之后使用全连接层降维进行分类。许多已经训练好的预训练模型，本质上已经具备了强力的提取浅层基础特征和深层较为抽象特征的能力，如果从头对于新的卷积神经网络完全使用自己的数据集进行训练，有两点缺陷。一是耗费大量的计算时间和计算资源。二是经验欠佳加训练数据的选取，容易导致模型准确率低，过拟合等诸多问题。本课题选用的就是在 ImageNet^[12]数据集上已经训练完成的 VGG19 网络进行微调。

3.3.2 什么情况下使用微调

微调在以下状况时建议使用。

(1)模型使用的训练集和预训练模型使用的训练集很相似，可以保证提取的基础特征和抽象特征也很相似。假如预训练集使用的是风景图片进行预训练，而微调使用的模型要对人脸进行分类，则效果一定欠佳。

(2)自己搭建的模型在经过训练后准确率一直很低。

(3)数据集的数据量太少，分布不合理。

(4)没有足够的硬件计算资源支持大规模权重训练。

3.3.3 微调的注意事项

(1)首先将预训练的参数装载进预训练的模型中得到原始模型。之后一般情况下会去掉输出层加入新的全连接层进行特征提取或者加入新的 softmax 分类层直接进行分类。

(2)由于装载的权重得到的训练结果已经非常不错，在训练时应选用较小的学习率进行训练，避免装载的权重值得到过大的扭曲反而降低准确率。

(3)如果新训练的数据量过小，直接修改最后一层进行分类即可，如果数据量中等偏大，可以将前几层的参数进行冻结训练最后几层中层特征。

3.3.4 微调与迁移学习的区别

微调只能看作是迁移学习的一个子集。迁移学习大体分为两个部分，一是对于知识的迁移，二是对于知识的再加工以便于对新的问题进行解决。如何将知识抽象出来和如

何再加工则是可以通过不同的组合得到的。微调则是将原模型的权重值用来对新的数据进行训练，这里迁移的知识是预训练好的权重，再加工是新的训练，所以微调只是一种具体的手段，而迁移学习则是一种更加抽象的思想指导我们进行快速的实验。

3.4 特征提取和检索

3.4.1 数据预处理

(1)图像大小统一。Caltech101 中的图像在二维上的像素点个数不近相同。对于卷积神经网络，输入的格式一定要是统一的，尤其是对于 VGG19 一般采用 (244, 244) 作为输入维度。

(2)输入归一化

图像在输入时要提前算出 RGB 三个维度的均值和方差，进行归一化，加快训练的速度。

(3)随机编码匹配

将 101 类图像的类别分别随机指定为 0 至 100 中的任何一个数字，同时将每一张图片的名称作为 key，其对应的类别数作为 value 组成 dict。再将每一个图片名称及其维度为 (224, 224, 3) 的 numpy 数组组成 tuple 作为一个元素装入 list。以上这两个对象在之后都会用来看做数据的再组装从而帮助分析实验结果。

(4)独热编码模式转换。在训练数据的标签匹配时，如果使用上述的整型数字进行匹配，很难做到了真正的特征互斥。而这恰恰是独热编码模式的优势。还有一个好处即是将特征稀疏化，原因是将之前使用十进制整型的数字投射到了 N 维的二值空间。

3.4.2 特征提取与检索

神经网络的诞生原本就是在模拟我们真正的人脑思维，我们可以在神经网络中清晰的判定浅层和深层的特征，越靠近输入层的特征是低层特征，是高维的，往往不适于作为特征提取，越靠近输出层，是高层特征，维度较低，适用于作为特征提取。往往在实际应用中，是将分类层的倒数第二层，即 4096 维的全连接层作为特征进行提取。

本课题参考 2015 年论文 Deep Learning of Binary Hash Codes for Fast Image Retrieval^[13] 在最后一层 4096 维的全连接层之后添加一层极少神经元的隐藏层命名为哈希层，再添加一层随机失活层防止过拟合，最后再添加分类层验证准确率。同时装载 VGG19 在 ImageNet 中已经训练好的权重参数，冻结除了新添加的最后三层进行微调得到最后三层的新权重参数。

尽管是简单的添加了一层隐藏层，但是这一层可以作为中层特征（4096 维的全连接层）和高层特征（输出层）的过渡，作为哈希函数的学习阶段。之后，将结果带入如下的哈希函数集中进行编码，即我们人为的对这一层进行二值的判定，由于以 0.5 作为阈值，将得到一串二进制数，这就是我们进行粗检索所依据的特征。

$$H^j = \begin{cases} 1 & Out^j \geq 0.5 \\ 0 & otherwise \end{cases} \quad \text{式(3-3)}$$

课题中我们使用的是具有 8 个神经元的隐藏层作为哈希层。如果以二进制所代表的十进制作为元素的标签，则 8 位的哈希值可以至多将图片大体分为 256 类，是稍大于 10^1 的。如果我们将图片给予一个新的标签，将其放在比人工标注的类别更多的“桶”中，相比于普通的利用 4096 维特征进行检索，根据设定阈值筛选排序的方式，先利用二值哈希快速的根据汉明距离筛选出最邻近的几个桶进行粗查找，之后在确定的几个桶中按原来的方式进行细查找，效率将会大大增加。

实际上，这种引入哈希值作为特征进行粗查找的方法不仅节约了时间，而且也降低了空间。因为哈希的低维度在数据库中存储是十分节省资源的，并且随着图片指数级的增长，类别的不断增加，引入哈希值特征的检索时间相较于普通的线性搜索将会是近对数与线性的关系。

3.4.3 Fine-tune 模型训练

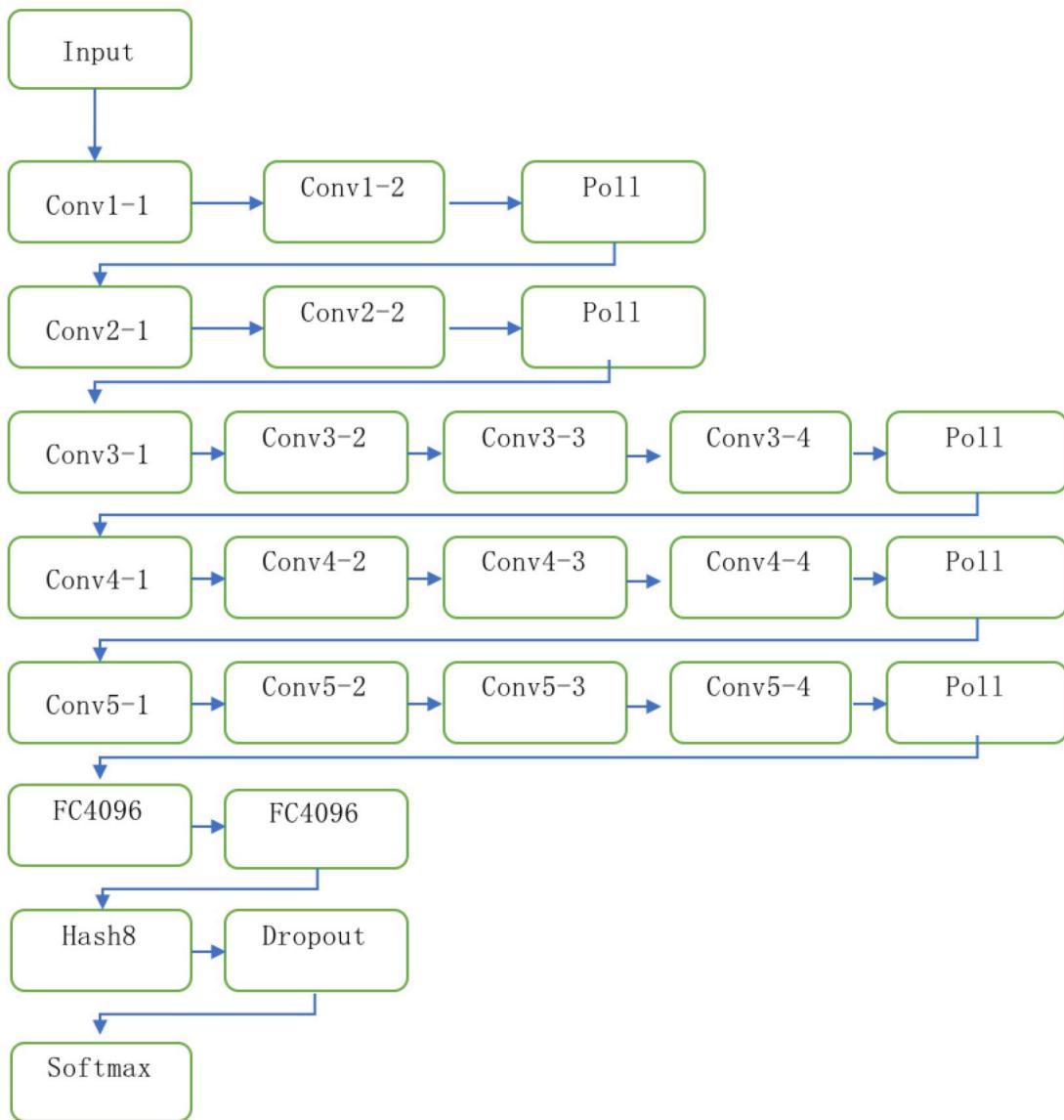


图 3-2 VGG19 fine-tune 模型

- (1)损失函数选用 binary_crossentropy
- (2)选择随机梯度下降作为反向传播优化算法
- (3)学习率选用微调的常用数字 1e-4，动量大小为 0.9。
- (4)批处理大小选用 32 以适应内存大小，训练批次只需 3 次，同时在训练中打印日志我们可以发现准确率一直保持在 99%左右并未降低，这就是微调的威力，而中间哈希层的中层特征已经被我们快速的提取出来。
- (5)训练的模型通过调用 Keras 的 model.to_json()函数导出为 json 文件存储，同时利用 save_weights 函数将新的 h5 权重文件存储。

3.5 本章小结

本章主要介绍了基于卷积神经网络的检索系统实现。首先简单介绍了卷积神经网络兴起的原因，卷积神经网络与普通神经网络的区别以及如何在训练时有效避免过拟合问题。之后对于卷积神经网络训练时的方法进行了总结，主要有数据归一化，批处理训练及对于图像特有的数据增强技术。之后对于本课题中选用的卷积神经网络结构进行了分析，并且给出了微调的网络结构和特征提取方式及其相似性度量方式。为后文当检索系统作为整体出现在服务端做准备说明。

第四章 检索系统的功能设计

4.1 系统的技术框架

本系统的功能主要有两种，库内检索和上传检索。如下是技术框架与流程图。

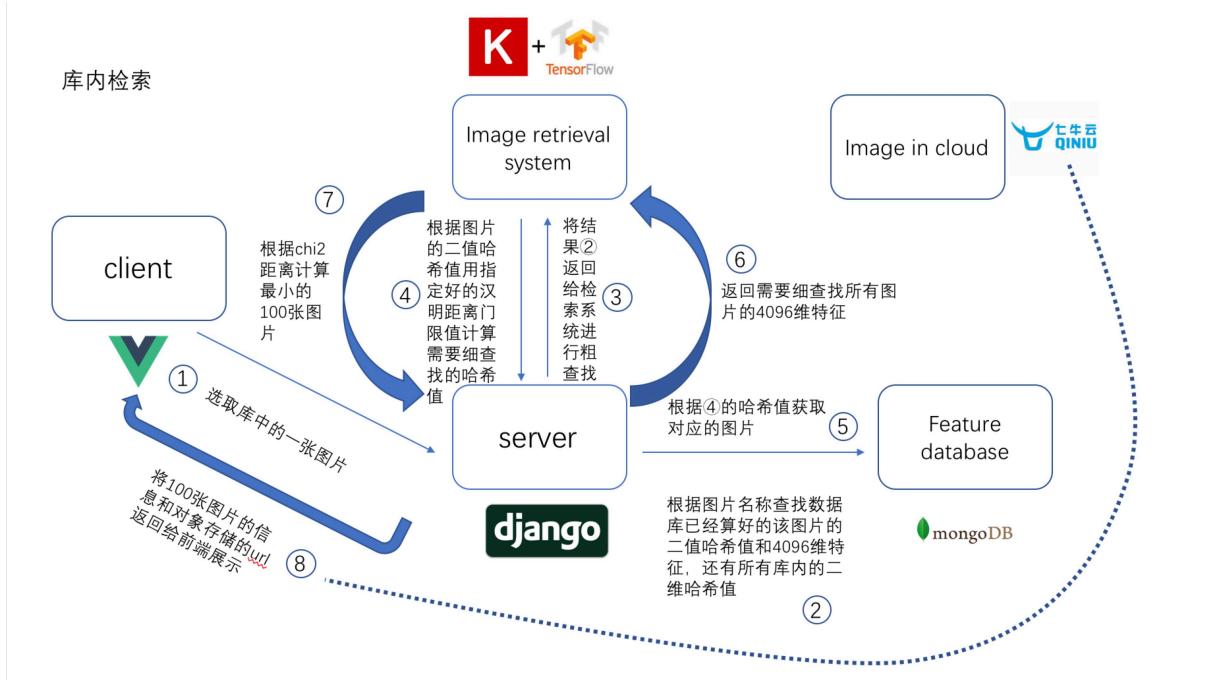


图 4-1 库内检索流程图

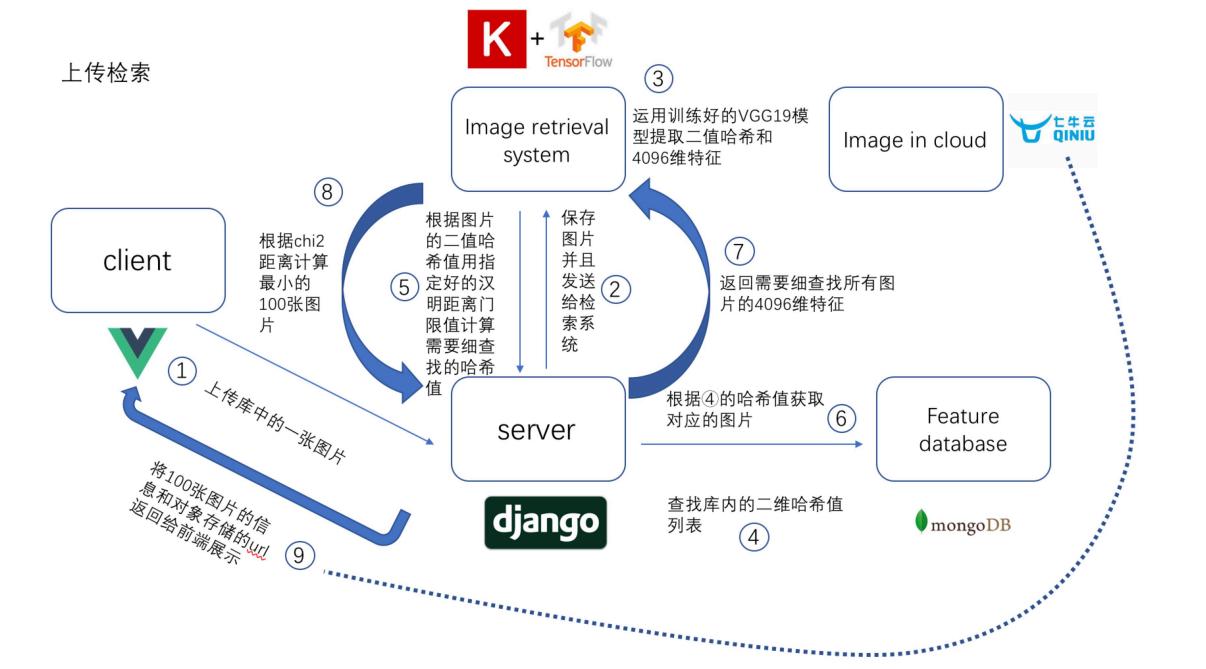


图 4-2 上传检索流程图

技术栈如图所示：前端为 Vue，后端为 Django，检索系统前端使用 Keras，后端设定为 TensorFlow，数据库采用 MongoDB 存储图片信息和特征，图片保存在七牛云上做静态对象存储。

4.2 系统的工作流程

界面在初始化后会从库中挑选一定量的图片返回，库中的所有图片的哈希特征和 4096 维特征均已存入 MongoDB 数据库中。

4.2.1 库内检索

交互模式：用户点击界面的任意一张图片或者点击随机选取按钮进行检索
流程：

① 用户从前端选取一张图片 ② 服务端根据 API 接口拿到图片的名称在数据库中进行查找该图片的 document，从而得到该图片之前训练得到的哈希特征和 4096 维特征，与此同时还要获取到数据库中所有的二值哈希列表。③ 将上述所有的结果返回给检索系统。④ 检索系统根据预设的汉明距离门限值进行粗检索，返回筛选后的哈希特征列表。⑤ 服务端根据二值哈希列表在数据库中查询符合条件的所有图片 ⑥ 将结果返回给检索系统进行细查找⑦ 检索系统根据卡方距离及其门限值进行细查找得到距离最小的 100 张图片返回给服务端。⑧ 服务端将检索结果返回给前端并且根据返回的图片 url 向七牛云获取静态资源。

4.2.2 上传检索

交互模式：用户点击上传按钮进行检索
流程：

① 用户从前端上传一张本地图片 ② 服务端通过 API 接口拿到图片并保存，再将图片发往检索系统③ 检索系统装载预训练好的 h5 权重文件对上传的图片进行特征提取得到新的哈希特征和 4096 维特征。④ 获取到数据库中所有的二值哈希列表。⑤ 检索系统根据预设的汉明距离门限值进行粗检索，返回筛选后的哈希特征列表。⑥ 服务端根据二值哈希列表在数据库中查询符合条件的所有图片⑦ 将结果返回给检索系统进行细查找⑧ 检索系统根据卡方距离及其门限值进行细查找得到距离最小的 100 张图片返回给服务端。⑨ 服务端将检索结果返回给前端并且根据返回的图片 url 向七牛云获取静态资源。

4.3 关键模块分析

4.3.1 客户端

(1) 技术栈简介：Vue。Vue 是现在最火热的一款用于快速构建用户界面的前端框架。其特殊之处在于能够让用户体验到渐进式的开发。Vue 既可以作为轻量级的插件融入到其他已经成熟的项目中，也可以作为主要开发模式开发大型单页面并且与其社区的丰富

插件整合实现各种交互和功能。由于本课题需要快速搭建一个简单交互的单页面应用并且与后端进行交互，Vue 就成为了我们的首要选择。

(2)功能：负责用户和后端系统的交互，交互模式有三种，前两种为库内检索，即点击任意一张图片或者点击随机从库中取图，最后一种为上传检索，即点击上传按钮从本地上传图片

(3)API 介绍

表 4-1 API 介绍

说明	URL	Method	Param
点击任意一张图	/cnn_vgg19	GET	name
随机取图	/cnn_vgg19_random	GET	-
上传图片	/cnn_vgg19	POST	-

4.3.2 服务端

(1)技术栈简介：Django。选用 Python 作为后端语言的原因很简单，一是作为脚本语言能够快速的搭建系统，二是可以方便的将检索系统整合到我们的后端。Django 作为 Python 语言下最火热的 MVC 框架，可以清晰的将模型(model)，视图(view)和控制器(controller)三者分离开来。在 Django 的初始化中，在 models 文件夹中我们需要本地对 MongoDB 进行连接，在 views.py 文件中我们需要定义路由映射从而实现前端的交互，在 utils 文件夹中整合业务逻辑和检索系统。将三块重要的实现类型分离开来。

(2)功能：负责分别与客户端，检索系统，数据库的三方数据通信。对于客户端，根据相关 API 确定后端路由将前端数据返回给检索系统或者直接进入数据库查找，对于检索系统，及时的查询数据库中的图片特征信息。

4.3.3 检索系统

检索系统主要由特征提取模块和搜索引擎组成。

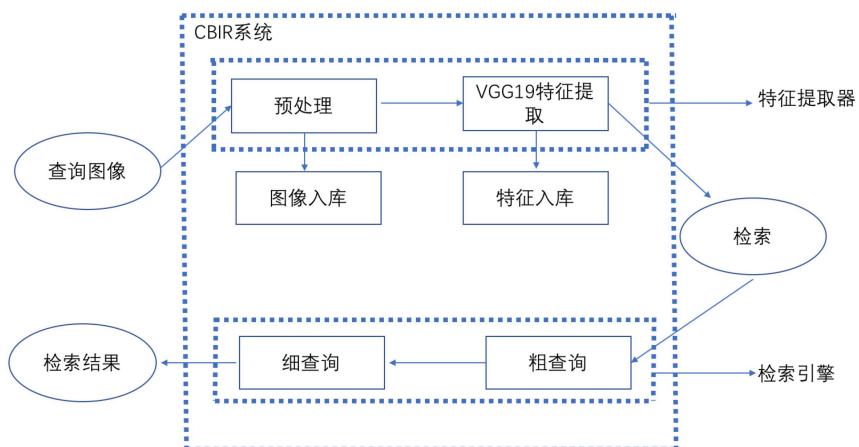


图 4-3 检索系统设计

(1)技术栈：前端 Keras 后端 Tensorflow

检索系统使用的是现如今最火热的深度学习 API 框架 Keras 作为前端与可选后端进行交互。这里我们选用的是 Tensorflow。Keras 的优势就是简单易用的 API 能够帮助研究人员快速将想法转换成实验结果。总的来说有三大优势，一是简单快速的模型设计，二是支持 CNN 和 RNN，三是支持 GPU。对于本课题还有一个重要优势就是 Keras 整合了最经典一些图像检索模型例如 VGG16,VGG19,ImageNet 等及其已经在大规模数据集上训练完成的 h5 权重文件，使我们使用 VGG19 进行 fine-tune 成为了可能。

(2)特征提取模块：

当前端用户选择上传图片时，图片首先会经过服务端到达特征提取模块，进行预处理，将 RGB 三个维度的像素值进行归一化。搜索引擎调取之前已经训练得到的 json 文件利用 Keras 的 `model_from_json()` 函数进行模型建立同时利用 `load_weights` 函数装载 h5 权重文件进行特征提取。

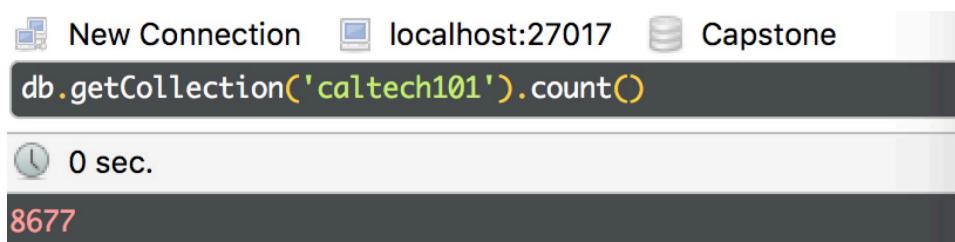
(3)搜索引擎模块：

粗筛选：搜索引擎会先根据待检索图片的 8 位二值哈希特征，以汉明距离为相似性度量，快速筛选出最邻近的几个类别

细筛选：搜索引擎再次从服务端获取到临近类别的所有图像根据卡方距离和设定阈值进行线性扫描排序获取最终结果

4.3.4 数据库

(1)技术栈：MongoDB。MongoDB 是现如今最火热的基于分布式文件存储的非关系型开源数据库。MongoDB 会将每条数据存储成一个文档（document），数据结构类似于 JSON 文件是以键值对的形式。字段的类型包括基础的整数，字符串，数组等，也支持 Binary 类型存储，方便我们将高维特征存储在数据库中。同时我们在 Django 服务端中引入了 MongoEngine。MongoEngine 本质上是一个对象文档映射器（ODM），同时也相当于一个基于 SQL 的对象关系映射器（ORM）。能够帮助我们在服务端使用 python 语言执行相关的 API 对数据库进行 CRUD 操作。



```
New Connection localhost:27017 Capstone
db.getCollection('caltech101').count()
0 sec.
8677
```

图 4-4 数据库图片总数

Key	Value	Type
(1) ObjectId("5af10d77985c13099d368b03")	{ 6 fields }	Object
_id	ObjectId("5af10d77985c13099d368b03")	ObjectId
feature	<binary>	Binary
hashcode	00000101	String
image_name	grand_piano/image_0095.jpg	String
isInDatabase	true	Boolean
intID	1	Int32

图 4-5 字段名称和类型

(2) 存储细节：

每一条记录在 MongoDB 中都作为一条 document

_id: MongoDB 为每一个 document 建立的唯一的哈希值索引

feature: 图片的 4096 维特征，利用 Binary 数据类型进行存储，在 python 中可以利用 pickle 包快速的复原成 numpy 数组

hashcode: 8 为哈希码字符串，用于粗检索

image_name: 库内图片的唯一标识，用于检索图片信息

isInDatabase: 判断该图片是否在 Caltech101 库内

intID: 由于 MongoDB 不支持随机查询，这里按入库顺序指定号码，用于随机检索

4.4 本章小结

本章首先通过示意图对于检索系统整体架构进行了分析，展示了数据的流通过程。之后对于每一步进行了细致的分析。包括前端的数据如何通过 API 接口发送往后端，后端在何时通知检索系统进行特征提取及相似性度量以及后端在何时查询数据库将结果返回给检索引擎。之后对所有模块的技术栈进行了分析及其选取原因。

第五章 检索系统的实现

5.1 系统的主界面

页面上方是系统名称，下方是检索的 100 张图片结果，每张图片均可点击。右方的三个按钮分别会显示一个说明弹框，进行随机库中检索，上传图片。

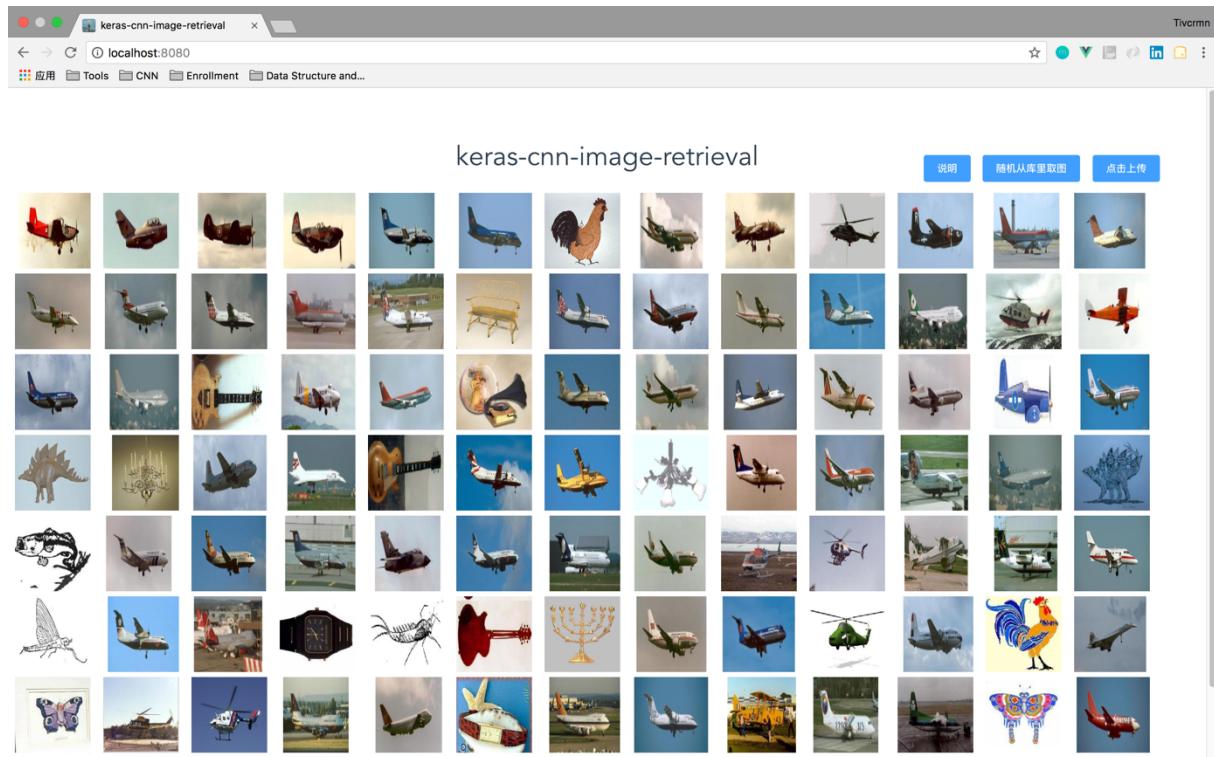


图 5-1 系统主界面

5.2 库内检索

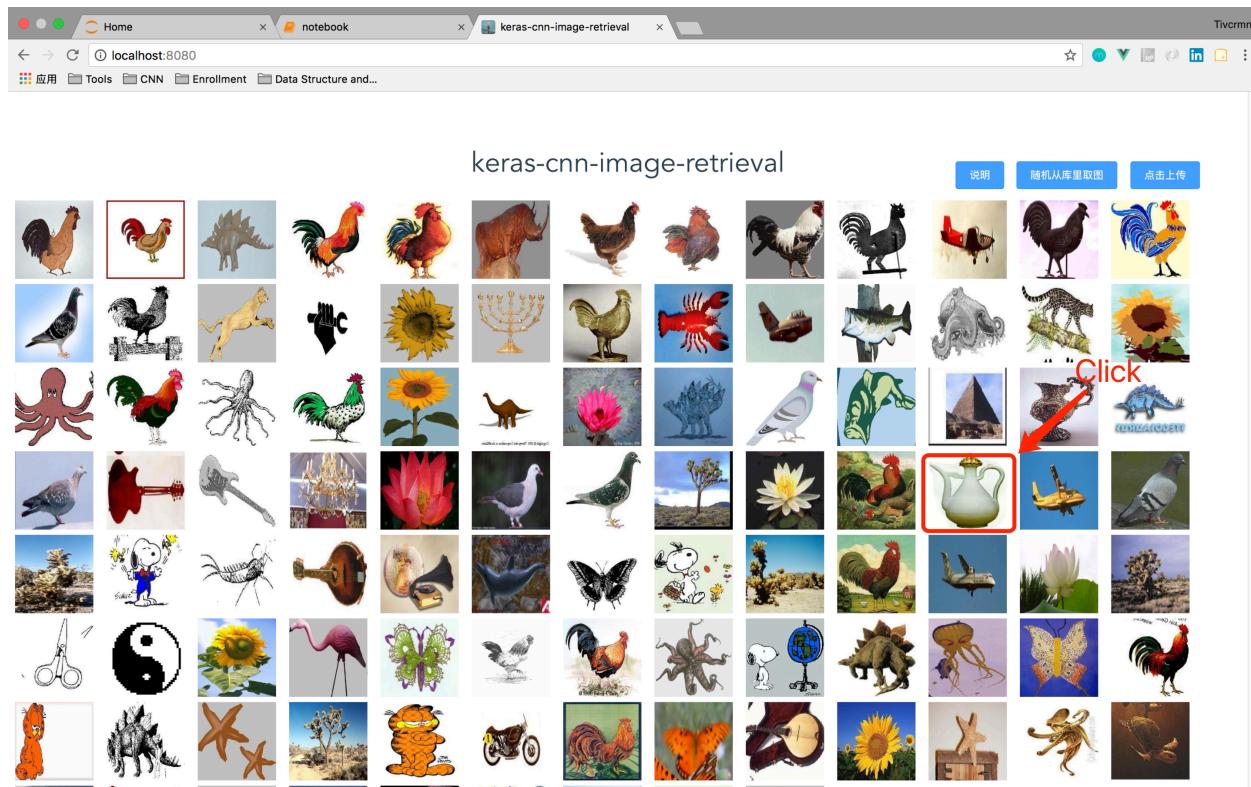


图 5-2 点击图片中的水壶

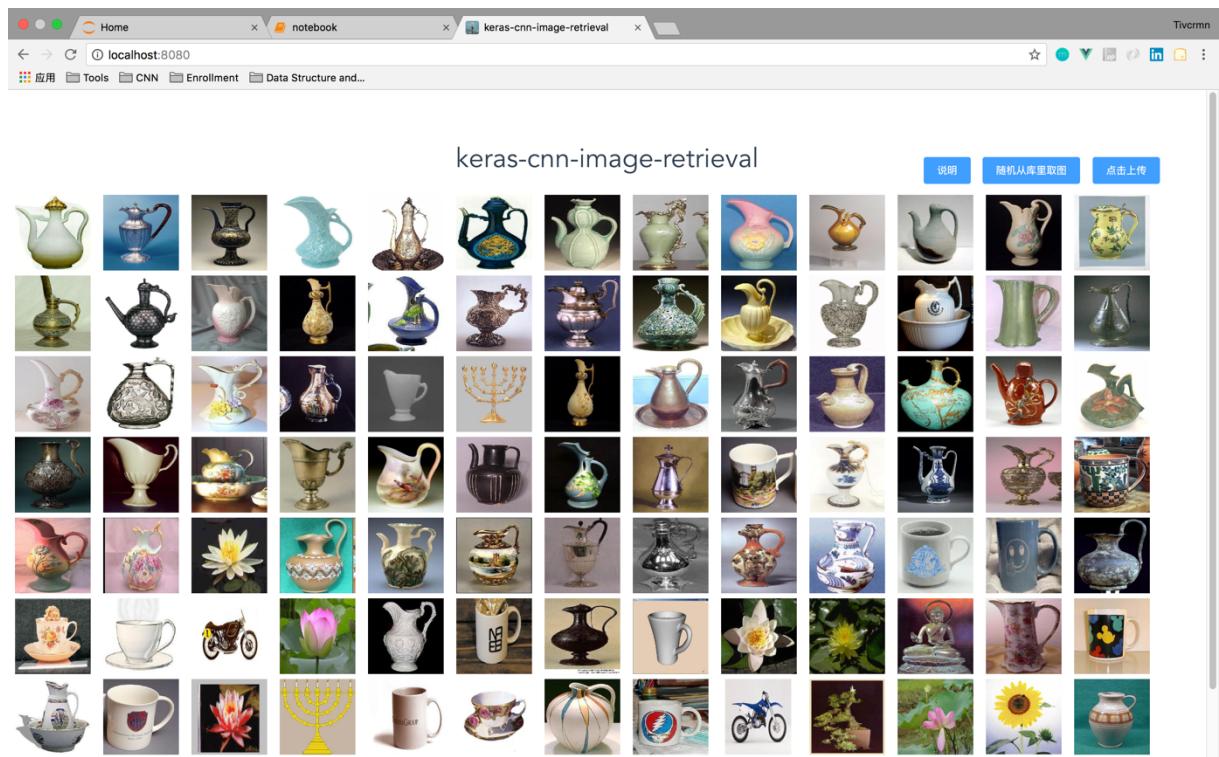


图 5-3 显示检索结果

5.3 上传检索

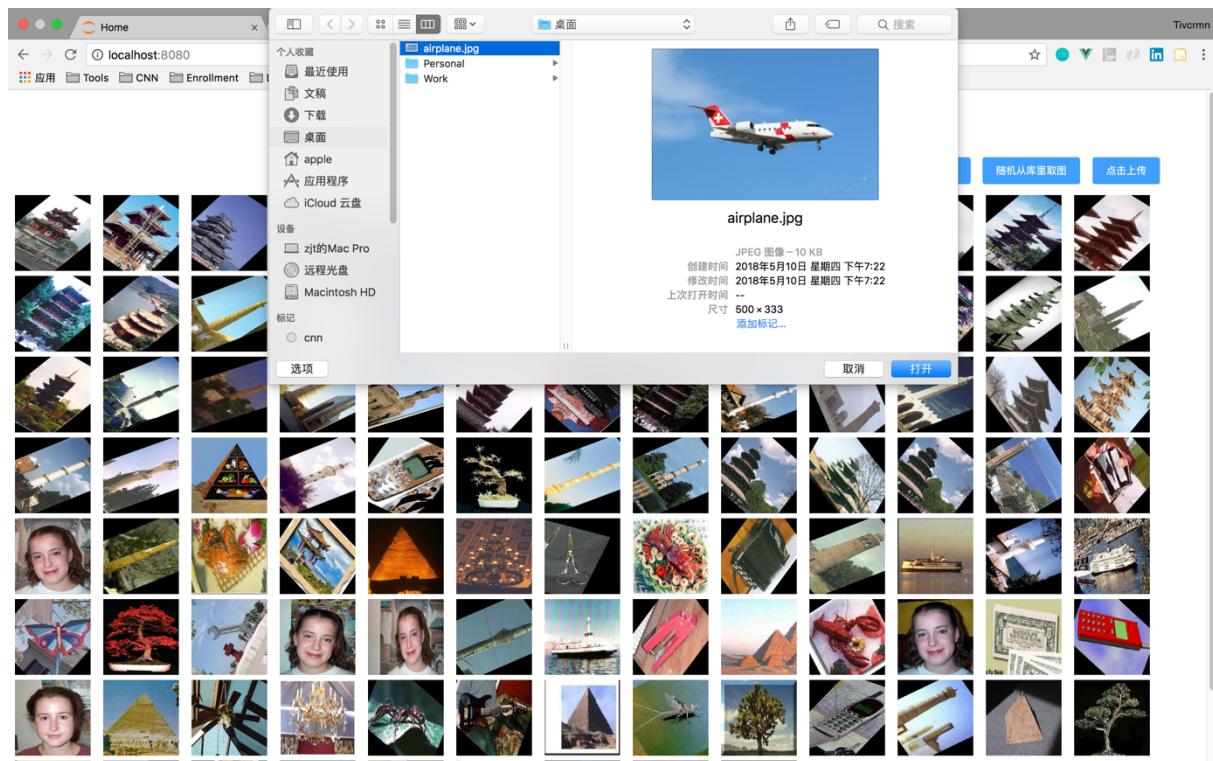


图 5-4 上传一张飞机的图片

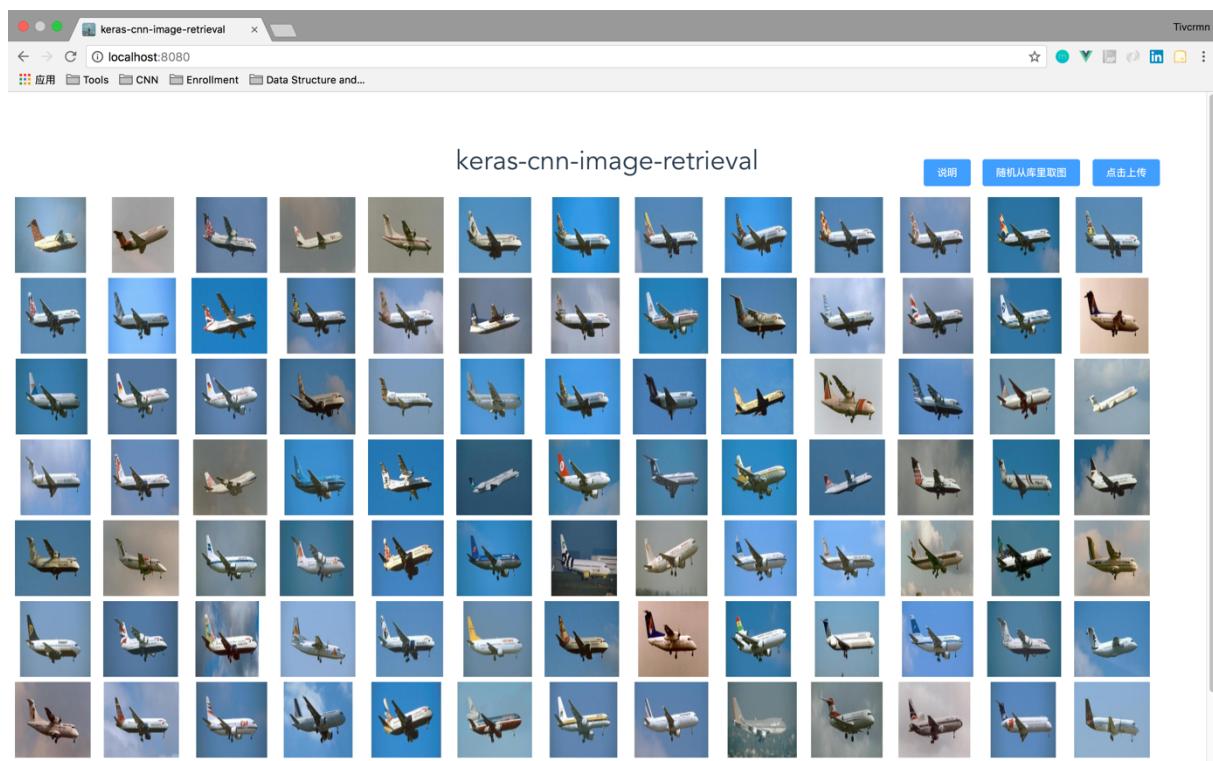


图 5-5 检索结果为飞机

5.4 系统结果分析

本节对比在 40 次库内随机检索中，基于哈希的检索和普通的检索的结果对比。

表 5-1 40 张待检索图片的类别和类内序号

序号	Image Class	Image Index	序号	Image Class	Image Index
1	saxophone	16	21	hawksbill	79
2	Faces_easy	21	22	Motorbikes	570
3	car_side	72	23	Faces_easy	294
4	Faces_easy	97	24	grand_piano	50
5	crayfish	48	25	pyramid	39
6	saxophone	25	26	Faces_easy	87
7	anchor	10	27	saxophone	35
8	umbrella	70	28	brontosaurus	12
9	soccer_ball	26	29	stapler	28
10	bonsai	34	30	brain	69
11	brontosaurus	13	31	llama	37
12	sunflower	72	32	gerenuk	23
13	Motorbikes	736	33	Motorbikes	797
14	Motorbikes	747	34	scorpion	28
15	ant	37	35	tick	33
16	airplanes	350	36	cougar_face	16
17	hedgehog	41	37	yin_yang	12
18	pigeon	22	38	panda	9
19	Motorbikes	537	39	airplanes	278
20	butterfly	82	40	ant	34

表 5-2 精确率

序号	哈希%	普通%	序号	哈希%	普通%
1	50	60	21	95	95
2	80	80	22	100	100
3	100	100	23	85	85
4	100	100	24	100	100
5	35	35	25	75	75
6	95	95	26	65	65
7	5	5	27	90	90
8	10	10	28	10	5
9	100	100	29	20	75
10	100	100	30	100	100
11	25	40	31	70	75
12	100	100	32	55	70
13	100	100	33	100	100
14	100	100	34	65	75
15	75	90	35	60	70
16	100	100	36	100	100
17	100	90	37	85	100
18	95	90	38	25	90
19	100	100	39	100	100
20	100	100	40	20	25

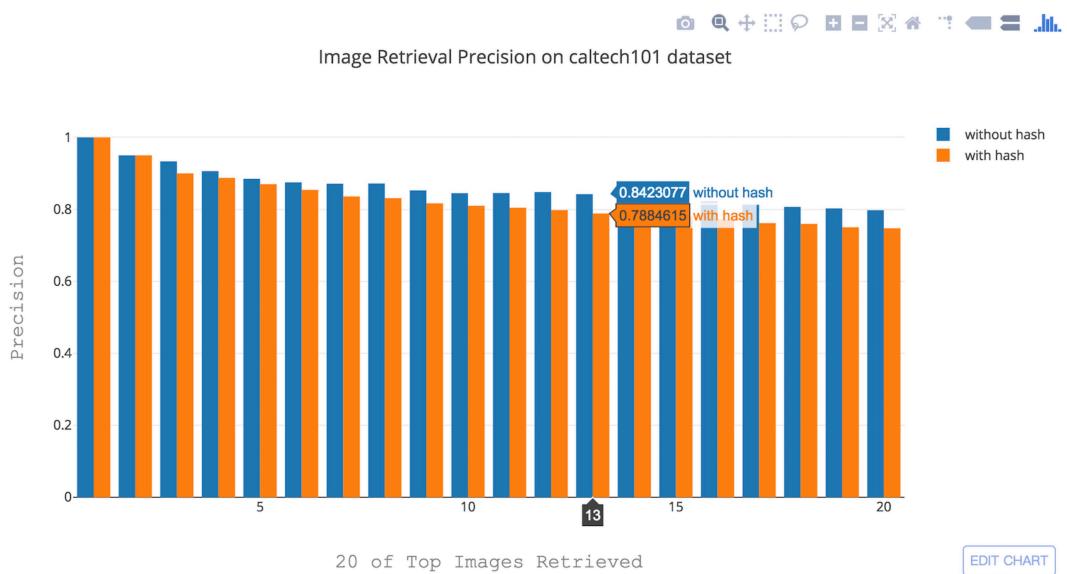


图 5-6 按照检索相似度排序分别在每一位的 MAP 平均精确率

从分析图中可以看出，引入哈希的检索在排序结果的每一位的精确率只是稍微略低于普通检索，随着排序位次的增加精确率的差值会变大。

表 5-3 时间对比

序号	哈希 s	普通 s	降低率%	序号	哈希 s	普通 s	降低率%
1	0.077	0.394	80.344	21	0.178	0.358	50.174
2	0.194	0.384	49.447	22	0.177	0.341	47.837
3	0.130	0.387	66.454	23	0.244	0.417	41.431
4	0.161	0.402	59.748	24	0.143	0.374	61.791
5	0.213	0.431	50.561	25	0.171	0.353	51.305
6	0.212	0.351	39.461	26	0.115	0.329	64.944
7	0.038	0.326	88.160	27	0.271	0.426	36.391
8	0.171	0.324	47.021	28	0.041	0.345	87.884
9	0.095	0.380	74.914	29	0.145	0.399	79.019
10	0.169	0.408	58.603	30	0.083	0.399	79.019
11	0.089	0.319	71.809	31	0.117	0.427	72.407
12	0.187	0.335	44.095	32	0.119	0.360	66.766
13	0.080	0.326	75.338	33	0.063	0.420	84.905
14	0.136	0.336	59.322	34	0.077	0.366	78.940
15	0.090	0.336	73.143	35	0.170	0.329	48.368
16	0.227	0.325	30.238	36	0.142	0.354	59.788
17	0.091	0.420	78.231	37	0.107	0.353	69.607
18	0.193	0.426	54.486	38	0.093	0.323	71.001
19	0.232	0.329	29.672	39	0.211	0.368	42.709
20	0.050	0.334	84.8377	40	0.092	0.392	76.428

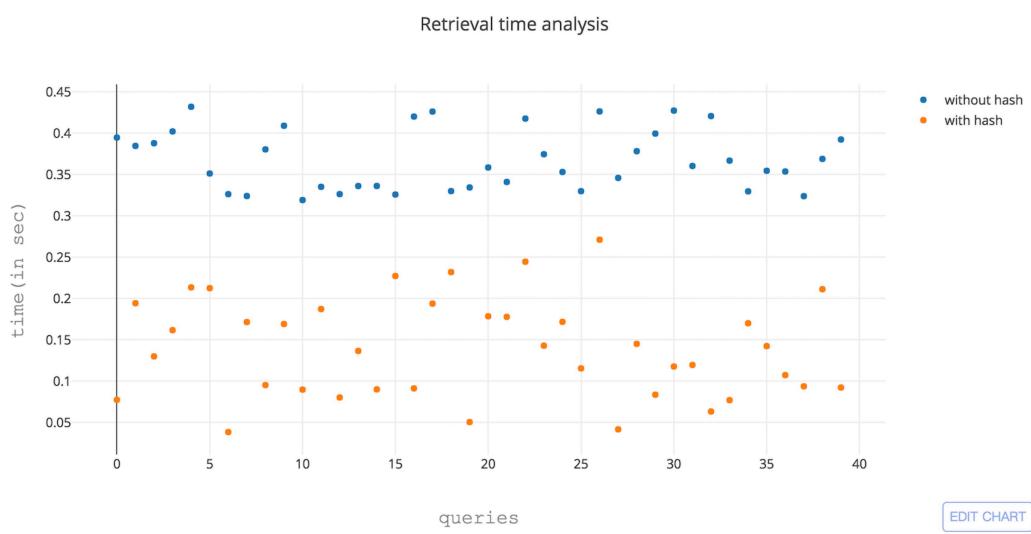


图 5-7 时间分布

在时间对比上，差异比较明显，降幅在 29%至 88%之间，但这只是在小训练集上，数据量和分类数目都不大，在越大规模的训练集上降幅会更加明显。

5.5 本章小结

本章对检索系统整体进行了展示，并且对于搜索引擎的效率，精确率和平均精度均值进行了分析。得出结论：引入哈希的两轮检索比单纯的一轮检索精确率和平均精度均值并不会降低太多，不仅如此，时间会大幅度降低。

第六章 总结和展望

6.1 总结

最初选择题目时因为对于现在火热的神经网络的好奇选择了这个题目，事实证明确实在这个过程中学习到了很多的算法知识，同时从一个完全不懂的小白成为了一个合格的入门者。作为信息工程的学生，图像的处理也是通信领域中的一个小分类，因为之前选修过图像处理这门课，其中的相关知识让我在新的领域中学习的更加迅速。

总的来说，本次课题主要实现了一个基于卷积神经网络的图像检索系统。主要两部分，一是架构，二是算法，这同样也是现在工业界中必不可少的两个部分。

在算法中，之前在各种网课平台中学习基础的神经网络算法，对于神经网络的原理，模型的建立，参数的意义，训练的方式，训练中的小技巧等进行了简单的学习，同时也对于 Python 中的语法糖，各种数学图形库，深度学习框架 Keras 的 API 都提前进行了熟悉，为之后的研究打下了基础。在完成基础的神经网络训练和特征提取之后，在学长的帮助下尝试提高，于是将二值哈希特征引入了检索系统中，令人欣喜的是成功的将检索的时间大大缩短。

之后，在检索系统的运行流程跑通之后开始着手处理系统整体的设计，即如何将检索系统融合进服务端同时依靠服务端与前端，数据库通信。

总的来说，系统的基本功能都已经实现，但仍有不足。比如在数据集的选取上可以选取数据量更大，分类数更多的数据集，在测试二值哈希时通过调整分类数大小或许能获得更加令人信服的结果。还有在算法上，可以有更多的创新，对于现如今的各种哈希算法可以进行更深一步的研究。或创新，或将现有的不同哈希算法做融合从而进一步提高检索系统的检索效率。

6.2 展望

图像检索在未来会变得越来越重要，这是毫无疑问的。这是一个覆盖几乎所有领域的问题，新的算法在未来将会受到更大数据量的挑战。对于本课题的系统，还是远远不够的。有以下几个方面可以进行新的研究：

- (1) 在普通的图像检索算法上，将图像检索按领域设计相应的算法，将图像检索落地解决实际领域问题。
- (2) 研究各种特征提取算法，进行组合，搭建特征提取和检索 pipeline。
- (3) 引入反馈系统做实时反馈，不断迭代使检索系统精确率不断提高。

参考文献

- [1] D G Lowe. Distinctive Image Features from Scale-Invariant Keypoints[J]. International Journal of Computer Vision, 2004, 60(2):91-110.
- [2] H Bay. SURF : Speed-Up Robust Features[C]// European Conference on Computer Vision. 2006.
- [3] F Perronnin, C Dance. Fisher Kernels on Visual Vocabularies for Image Categorization[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2007:1-8.
- [4] N Dalal, B Triggs. Histograms of oriented gradients for human detection[C]// IEEE Computer Society Conference on Computer Vision & Pattern Recognition. IEEE Computer Society, 2005:886-893.
- [5] T Ojala, Pietik, M Inen 等. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns[M]// Computer Vision - ECCV 2000. Springer Berlin Heidelberg, 2000:404-420.
- [6] E Rublee, V Rabaud, K Konolige 等. ORB: An efficient alternative to SIFT or SURF[C]// IEEE International Conference on Computer Vision. IEEE, 2012:2564-2571.
- [7] H Jegou, M Douze, C Schmid 等. Aggregating local descriptors into a compact image representation[C]// Computer Vision and Pattern Recognition. IEEE, 2010:3304-3311.
- [8] G Csurka. Visual categorization with bags of keypoints[J]. Workshop on Statistical Learning in Computer Vision Eccv, 2004, 44(247):1--22.
- [9] N Srivastava, G Hinton, A Krizhevsky 等. Dropout: a simple way to prevent neural networks from overfitting[J]. Journal of Machine Learning Research, 2014, 15(1):1929-1958.
- [10] S Ioffe, C Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift[J]. 2015:448-456.
- [11] K Simonyan, A Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer Science, 2014.
- [12] J Deng, W Dong, R Socher 等. ImageNet: A large-scale hierarchical image database[C]// Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009:248-255.
- [13] Lin K, Yang H F, Hsiao J H 等. Deep learning of binary hash codes for fast image retrieval[C]// IEEE Conference on Computer Vision and Pattern Recognition Workshops. IEEE Computer Society, 2015:27-35.
- [14] A Babenko A, A Slesarev, A Chigorin 等. Neural Codes for Image Retrieval[J]. 2014, 8689:584-599.
- [15] 郑泽宇, 顾思宇. Tensorflow: 实战 Google 深度学习框架[M]. 电子工业出版社, 2017
- [16] 周志华. 机器学习 := Machine learning[M]. 清华大学出版社, 2016.
- [17] 黄文坚, 唐源. TensorFlow 实战[M]. 电子工业出版社, 2017

致 谢

非常感谢张彬老师在我本科的最终阶段给予我的指导，感谢您不辞辛劳地指导我完成毕设内容和修改论文内容。还要感谢实验室的陈超逸学长，你们在我完成毕设的过程中给予我非常大的帮助。最后要感谢我的母校，北京邮电大学，能在这里成长我感觉非常骄傲。