南 阳 理 工 学 院

本科生毕业设计(论文)

学院(系)： 软件学院

专 业： 网络工程

学 生： 陈浩

指导教师： 王浩

完成日期 2018 年 05 月

南阳理工学院本科生毕业设计（论文）

基于BOW的位置识别系统的设计与实现

Design and Implementation of   
Location Identification System Based on BOW

总 计：毕业设计(论文) 33页

表 格： 15个

图 片： 21个

南 阳 理 工 学 院 本 科 毕 业 设 计(论文)

基于BOW的位置识别系统的设计与实现

Design and Implementation of   
Location Identification System Based on BOW

学 院(系)： 软件学院

专 业： 网络工程

学 生 姓 名： 陈浩

学 号： 1415935157

指导教师(职称)： 王浩 讲师

评 阅 教 师： 王浩

完 成 日 期： 2018年05月02日

南阳理工学院

Nanyang Institute of Technology

基于BOW的位置识别系统的设计与实现

网络工程 陈浩

1. 位置识别是指当人在某个位置时获取当前位置信息的定位技术。由于美国的GPS全球卫星定位系统和中国的北斗卫星导航系统的普及，目前的定位技术已经非常成熟。但是民用级别的定位精度未能达到人们的日常需求，所以需要一种准确性较高且成本低廉的位置识别系统。本系统的架构为前端使用Keras深度学习框架配置VGG16预训练模型，后端使用TensorFlow深度学习框架提取图像的特征值，使用HDF5存储图像的特征值。本系统利用深度学习技术中的CNN（Convolutional Neural Network，卷积神经网络）为基础的VGG16预训练模型进行图像特征提取，根据对应的图像特征值即可搜索数据库中相似度最高的图像，通过用户的身边的图像识别用户的准确位置。本系统不但能进行位置识别，还能匹配图像的相似度，可对图像进行相似度搜索。由于Python编程语言的语法简洁和可读性强的特点，所以本系统优先选择Python编程语言。
2. 深度学习；CNN；位置识别；图像检索

Design and Implementation of   
Location Identification System Based on BOW

Network Engineering Major Chen Hao

1. Location identification refers to positioning technology that obtains current location information when a person is in a certain location. Due to the popularity of the US GPS global satellite positioning system and China's Beidou satellite navigation system, the current positioning technology is already very mature. However, the positioning accuracy of the civil level fails to meet people's daily needs, so a position recognition system with high accuracy and low cost is needed. The architecture of this system is to use the Keras deep learning framework to configure the VGG16 pre-training model at the front end, the TensorFlow deep learning framework at the back end to extract the feature values ​​of the image, and the HDF5 to store the feature values ​​of the image. The system uses the VGG16 pre-training model based on CNN (Convolutional Neural Network) in deep learning technology to extract image features. According to the corresponding image feature values, the image with the highest degree of similarity in the database can be searched. The image of the side identifies the exact location of the user. The system can not only identify the location but also match the similarity of the image, and can perform similarity search on the image. Because of the concise and readable nature of the syntax of the Python programming language, the Python programming language was chosen to implement the system.
2. [Deep](javascript:;) [Learning](javascript:;);CNN; Location Identification; Image Retrieval

目 录

[1 绪论 1](#_Toc513885801)

[1.1 背景 1](#_Toc513885802)

[1.2 研究现状分析 1](#_Toc513885803)

[1.2.1 室外定位技术 1](#_Toc513885804)

[1.2.2 室外位置识别技术 2](#_Toc513885805)

[1.3 实验内容 2](#_Toc513885806)

[1.4 实验目标 2](#_Toc513885807)

[1.5 论文总体结构 2](#_Toc513885808)

[2 基于深度学习的图像特征提取技术 3](#_Toc513885809)

[2.1 卷积神经网络介绍 3](#_Toc513885810)

[2.2 VGG模型介绍 4](#_Toc513885811)

[2.3 图像特征提取介绍 5](#_Toc513885812)

[3 可行性分析 5](#_Toc513885813)

[3.1 理论可行性 5](#_Toc513885814)

[3.2 技术可行性 5](#_Toc513885815)

[3.3 经济可行性 5](#_Toc513885816)

[4 系统设计与实现 6](#_Toc513885817)

[4.1 开发环境介绍 6](#_Toc513885818)

[4.2 系统总体设计 6](#_Toc513885819)

[4.3 特征提取模块 8](#_Toc513885820)

[4.4 特征数据库模块 11](#_Toc513885821)

[4.5 特征检索模块 13](#_Toc513885822)

[5 系统测试与分析 15](#_Toc513885823)

[5.1 实验数据集 15](#_Toc513885824)

[5.2 测试环境 15](#_Toc513885825)

[5.3 功能测试 16](#_Toc513885826)

[5.3.1 图像特征提取功能 16](#_Toc513885827)

[5.3.2 图像特征数据库功能 17](#_Toc513885828)

[5.3.3 图像特征检索功能 20](#_Toc513885829)

[5.4 性能测试 24](#_Toc513885830)

[5.4.1 特征提取测试 24](#_Toc513885831)

[5.4.2 特征检索测试 24](#_Toc513885832)

[5.4.3 特征检索准确率测试 25](#_Toc513885833)

[5.4.4 特征检索召回率测试 26](#_Toc513885834)

[5.5 测试结果分析 27](#_Toc513885835)

[结束语 30](#_Toc513885836)

[参考文献 31](#_Toc513885837)

[附录 32](#_Toc513885838)

[致谢 33](#_Toc513885839)

# 绪论

随着国家国防需求和人民出行需求的提升，位置识别精度需要更精准。现在的位置定位服务，如全球卫星定位系统和北斗卫星导航系统的军用已经可以满足国防需要。但是民用的精度和稳定性还是达不到人们的日常出行需求，而且目前普遍的定位系统都不能识别定位所在的地点名称，需要依赖于地图的功能。而定位系统在地图上存在误差，不能准确识别用户在某个建筑的具体方向。尤其是目前的定位系统在室内的定位精度会大幅降低，地下建筑影响最为严重。目前的定位系统在地下建筑基本失去作用，因为地下建筑信号差，无法与卫星进行正常通信。

本系统解决区域内定位精度低的问题，系统需要一定量的标志性且有意义的图像数据库。本系统能识别用户所在位置的具体信息，如某楼某方向。本系统推荐应用于局部区域，如学校、医院、商场、小区等区域性强的地方。

## 背景

在定位技术主要是卫星定位和移动通信基站定位的时代背景下，由于卫星定位的多种不确定性和高耗能，不能完美的应付当前移动互联时代。卫星定位对使用的环境较为苛刻，需要能和用户交互的卫星数量达到一定的数目，需要有良好使用天气，如晴天。还有卫星定位的室内定位精度极差，不能满足用户的定位需求。基站辅助定位技术的局限性大，只能用于允许定位精度误差较大的定位，如定位城市，定位小区。

在移动互联网高度兴起的时代背景下，急需一种对环境要求良好的定位技术，位置识别技术，能应用于室内室外的高精度定位。该技术可在无网络的情况下使用，不受天气的影响。该技术可应用于机器人位置识别、以用户为中心的位置识别。

## 研究现状分析

现有国内外相关研究，可以分为室外定位技术和室外位置识别技术两类。

### 室外定位技术

目前的室外定位技术主要分为卫星定位和移动通信基站定位两种技术：

基于卫星定位：

卫星定位是最常见的室外定位技术，但是卫星定位对于智能手机是极其耗电的。因此，一种常见的方法是周期性地使用。然而，卫星定位的责任循环对低能量的定位精度。卫星定位在城市地区通常不那么准确，误差较大，尤其是室内或者阴天。

基于移动通信基站定位：

移动通信基站定位是根据移动通信网络中的移动台定位，这种技术需要能连接移动通信网络的情况下，根据基站发射的信号到已注册的地点，即可知道定位所在的位置。这种技术只能定位到某个基站下的某个扇区，它的定位精度取决于基站的密集程度。一般误差大于200m，这种定位技术为精度最低的定位技术，只能用于已有定位技术下的辅助定位[21]。

### 室外位置识别技术

室外位置识别技术现一般采用计算机视觉领域的图像检索方法，有很多关于位置识别的研究，都是主要采用基于内容的图像搜索方法。Schindler 等人提出一种词汇树搜索算法实现在城市规模范围内进行目标位置识别。Chen 等人通过融合街道图像数据的两种传统表示方式，并改善图像中低对比度部分的特征检测方法，提高了对于地标建筑的识别性能。Guan 等人提出了一种去除无效特征，压缩图像特征向量的方法，使得基于图像的目标位置识别可以在移动终端上离线进行。王丽君等人经过实验得出利用卷积神经网络识别机器人视觉图像从而识别机器人所在位置，可在动态环境中准确识别位置。

近年由于深度学习的兴起和神经网络分类器准确率的提升，卷积神经网络逐渐被熟知，利用卷积神经网络进行图像检索从而识别位置成为可能，随着卷积神经网络的层数加深识别准确率也有所提升。

## 实验内容

首先把带位置标签信息的图像经过一系列预处理和使用卷积神经网络提取出图像特征，生成图像数据库，每个图像特征向量都对应一个位置信息标签。然后根据用户提供的含有位置信息的图像，使用同样的方法提取特征，把提取到的特征和图像数据库对比，取出最相似的图像，获取最相似的图像的位置信息标签，返回位置信息给用户。从而帮助用户通过提供位置信息图像，实现位置识别。

## 实验目标

根据用户提供的位置信息图像匹配图像数据库中最相似的图像，返回最相似图像的位置信息给用户，从而达到位置识别。

位置识别准确率在90%以上，识别说消耗的时间控制在3s以内，保证用户体验。

位置识别召回率与准确率需要达到某种平衡，当其准确率高时，召回率低，准确率低时，召回率高。

## 论文总体结构

本文章节结构如下：

第一章，绪论。本章节介绍了基于深度学习的图像搜索的目标位置识别方法的研究背景及其意义，然后分析了现阶段国内外学者们在目标位置识别方法上的研究现状，以及现阶段依然存在的问题，并且进一步提出本文课题实验目标。

第二章，基于深度学习的图像特征提取技术。本章介绍了本文实验工作中所涉及到的主要技术，及其简介。

第三章，可行性分析。本章介绍了本文实验的理论可行性和技术可行性，经济可行性等诸多可行性因素。

第四章，系统设计与实现。本章介绍了系统所依赖的开发环境，然后是对系统各功能模块的设计，最后对系统进行实现。

第五章，系统测试及性能分析。本章介绍了系统测试所需的数据集，和要做的工作。然后对实现的系统功能及方法性能进行测试，收集实验所得数据，统计实验数据结果，并且进一步分析系统的性能。

# 基于深度学习的图像特征提取技术

## 卷积神经网络介绍

卷积神经网络（CNN）是MLP的生物启发变体。从Hubel和Wiesel在猫的视觉皮层的早期工作中，我们知道视觉皮层包含复杂的细胞排列。这些细胞对视野的小部分区域敏感，称为接受区域。分区域平铺以覆盖整个视野。这些细胞充当输入空间上的局部滤波器，非常适合利用自然图像中强大的空间局部相关性[9]。

在[机器学习中](https://en.wikipedia.org/wiki/Machine_learning" \o "机器学习)，卷积神经网络（CNN或ConvNet）是一类深度[前馈](https://en.wikipedia.org/wiki/Feedforward_neural_network" \o "前馈神经网络) [人工神经网络](https://en.wikipedia.org/wiki/Artificial_neural_network)，已成功应用于分析视觉图像。

CNN使用设计为需要最少[预处理](https://en.wikipedia.org/wiki/Data_pre-processing" \o "数据预处理)的[多层感知器](https://en.wikipedia.org/wiki/Multilayer_perceptron)的变体。它们也被称为移位不变或空间不变式人工神经网络（SIANN），基于它们的共享权重结构和[平移不变性](https://en.wikipedia.org/wiki/Translation_invariance" \o "平移不变)特征。

与其他[图像分类算法](https://en.wikipedia.org/wiki/Image_classification" \o "图像分类)相比，CNN使用相对较少的预处理。这意味着网络学习传统算法中的[过滤器](https://en.wikipedia.org/wiki/Filter_(signal_processing)" \o "滤波器（信号处理）)是[手工设计的](https://en.wikipedia.org/wiki/Feature_engineering)。这种独立于先前知识和人力设计的特点是一大优势。

CNN由一个输入层和一个输出层组成，以及多个[隐藏层](https://en.wikipedia.org/wiki/Multilayer_perceptron" \l "Layers" \o "多层感知器)。CNN的隐藏层通常由卷积层，池层，完全连接层和规范化层组成。

目前主流的卷积神经网络架构有LeNet5、AlexNet、VGG、GoogLeNet、Inception V3、ResNet、SqueezeNet、等架构，这些架构代表了里程碑式的卷积神经网络的成就。

深度学习中，卷积神经网络代表着计算机的“眼睛”，能帮机器理解视觉，让机器能分析视觉图像。从而能进一步分析，产生计算机“意识”。

卷积神经网络过程如图2-1所示：

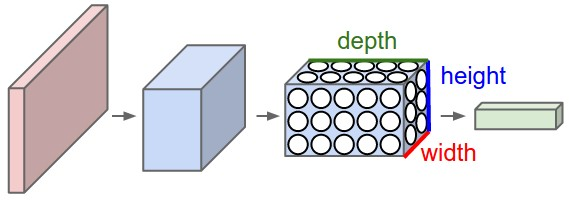


图2-1 卷积神经网络过程例图

## VGG模型介绍

卷积神经网络的配置（在列中显示）。配置的深度从左边（A）增加到右边（E），因为添加了更多的层（添加的层以粗体显示）。卷积层参数被表示为“conv接收域的信道数”。为了简洁，没有显示ReLU激活函数[8]。

VGG模型结构如图2-2所示：

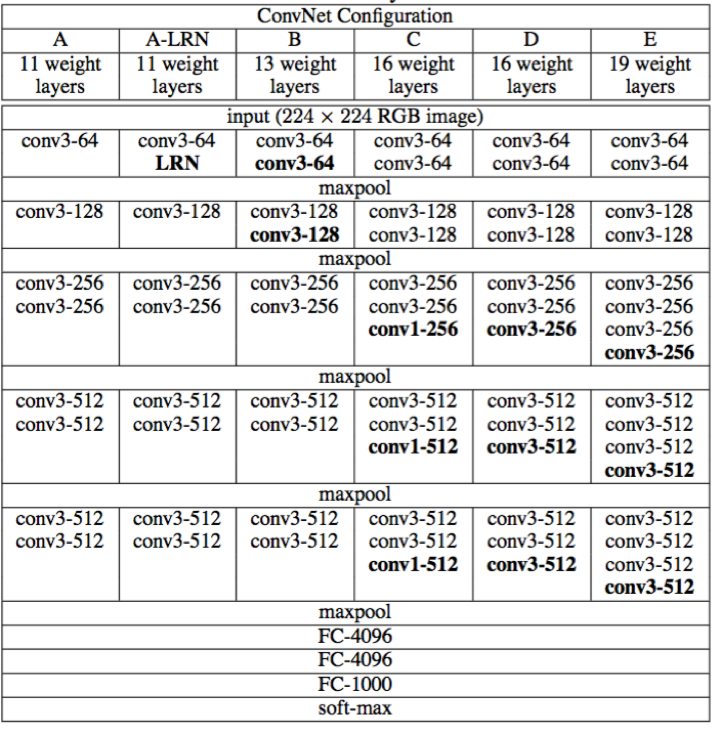


图2-2 VGG模型结构图

本次系统采用VGG16模型，使用图2-2中的D结构。D结构有五个池化层，五个卷积部分。卷积有着同样的filter，步长为1，填充为1，尺寸为3x3。VGG16有着16层，十三层卷积层，三层全连接层。VGG模型的FC全连接层都进行dropout正则化，提升模型的抗过拟合能力。

## 图像特征提取介绍

图像检索过程中，核心技术就是特征值提取，提取方法直接影响图像检索系统准确率。图像检索一般包括图像预处理、图像特征提取、图像特征检索三个方面，其中图像特征提取是关键一步。本次实验使用深度学习的卷积神经网络方法进行特征值提取，神经网络层数为16层。使用Keras深度学习框架的预处理模型对输出图像进行预测，无需重新训练神经网络，节约性能开支。Keras后端使用TensorFlow深度学习框架，TensorFlow为由C++编写的高性能深度学习框架，能够处理大型浮点运算。

# 可行性分析

## 理论可行性

现深度学习已经逐渐成熟，尤其是图像识别领域。深度学习图书近两年保持着高增长，相关书籍为深度学习相关方向的人提供技术支持。尤其是深度学习框架的诞生，使其深度学习逐渐简单化，使用方便，实现功能的代码量少。

深度学习下的卷积神经网络为图像识别领域的核心技术，犹如为图像识别领域量身定制的技术。卷积神经网络使其图像识别领域重要比赛ImageNet的图像识别准确率大幅度提升。2015年，微软的ResNet成功训练了152层深的神经网络，获得2015年的ILSVRC比赛的冠军，top-5错误率降低至3.46%。由此可见，图像识别越来越依赖于卷积神经网络。

## 技术可行性

以Keras为深度学习前端框架，TensorFlow为深度学习后端框架的图像识别系统现如今以应用广泛，可以以最少的代码开发出强大的深度学习系统。

HDF5存储功能支持线性代数中的矩阵和向量的存储，HDF5是特征数据库最好的选择，完整的保留了原数据格式和数据类型。HDF5使用文件存储，直接对文件进行操作，读取速度快，稳定性强。

Python作为一门高级解释型语言，适用于人工智能领域的开发，为该系统提供了重要的支持，减少了复杂的代码，支持使用大量的三方库，减少系统代码逻辑。

本次实验前，阅读了大量的文献和书籍，为实验提供技术储备。

## 经济可行性

本系统在个人笔记本上进行开发，开发使用的软件无需付费，开发场地为学校社团，经济上成本几乎为零。

Keras和TensorFlow为免费软件，Python为开源软件，技术上遇到的问题不需要付费资讯，向导师提问或者在参考相关书籍和网络博文即可解决，技术上无经济支出。本系统为免费试用，代码开源、共享，为开源软件做出贡献。

# 系统设计与实现

## 开发环境介绍

开发过程中，搭建开发环境是该系统开发的起步过程，同时标志着该实验开始的标志，是非常重要的一步，这一部分重点讲解如何配置开发使用的环境。

本次操作系统

本次系统开发过程使用的开发环境如表4-1所示：

表4-1 开发环境表

| 开发环境 | 操作系统：Microsoft windows 10 enterprise 64Bit |
| --- | --- |
| 编程语言：Python for Windows 3.6.4 64Bit |
| 开发IDE： PyCharm 2018.1 professional edition 64Bit |
| 三方库： Numpy、HDF5、Matplotlib、Argparse、Keras、PIL、OpenCV2、TensorFlow、Time |

## 系统总体设计

系统设计为三大功能模块：

1、图像特征提取模块

2、图像特征数据库模块

3、图像特征检索模块

系统流程为，系统接收到的原始图像经过图像特征提取模块时，进行图像预处理，然后提取特征值，最后进行特征值归一化处理。经过图像特征提模块得到的最终特征值，才能应用于特征数据库或者特征检索。

如果应用在特征数据库需要提供图像特征值和图像位置信息，才能录入数据库。录入数据库是需要将图像特征值和图像名称建立索引，最后存入特征数据库，写入数据库文件。

如果应用在检索系统，则需要读取特征数据库中的所有信息，将读取的信息和图像特征值进行匹配。然后排序，按照相似度从高到低排序，生成排序列表，最后输出检索结果图像名称，再根据图像名称查得图像信息。

系统架构流程图如下图4-1所示：

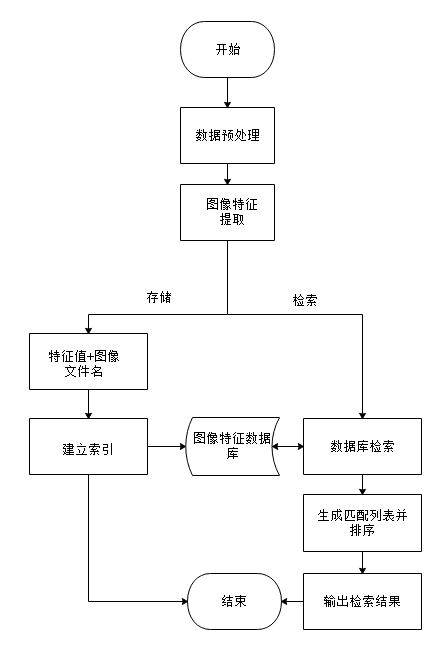


图4-1 系统总体流程图

根据系统总体架构流程图4-1，设计系统目录结构。系统目录结构为，根目录下存在三个子目录，特征数据库目录、系统日志目录、测试图像目录。其余的为相关系统程序，主要程序有特征提取程序、特征数据库程序、特征检索程序,最后为系统所需要的基础库程序.基础库程序提供系统最基本的数据处理功能，如带路径的文件名去除其路径，文件名去除文件类型后缀，文件名添加类型后缀等等功能。

特征数据库目录中的imageCNNAll.h5文件为特征数据库文件，内部存储了所有图像训练集图像的特征向量和图像名称。

日志目录中的image\_link\_info.txt文件为图像训练数据集的图像名称和图像对应的位置信息。

一级目录中的imagetest目录为系统测试图像所在目录，提供测试图像给本系统进行相关测试。

系统目录结构如表4-2所示:

表4-2 系统目录结构图

| 根目录 | 一级目录 | 二级目录 |
| --- | --- | --- |
| imageSearch/ | base.py | 无 |
| feature\_extraction.py | 无 |
| imageQuery.py | 无 |
| feat2file.py | 无 |
| models/ | image\_Feature\_Test.h5  imageCNNAll.h5 |
| log/ | image2info.txt  image\_link\_info.txt |
| imagetest/ | 19700102125648863.JPEG  19700102133705216.JPEG  19700102134147686.JPEG |

## 特征提取模块

如章节1.3所述，本文通过基于深度学习实现位置识别。特征提取是本系统最重要的模块，特征提取的方法使用深度学习的卷积神经网络方法。卷积神经网络有许多预训练模型，如Xception、VGG16、VGG19、ResNet50、InceptionV3、IncetionResNetV2、MobileNet等等预先定义好的卷积神经网络模型。而这些模型很容易的使用，预训练模型中的超参已经被调至较优状态，无需干预，只需直接利用即可。

以下为Keras框架提供的预训练模型表，其中详细的列举各种预训练模型的常见参数。

表4-3 Keras预训练模型性能表

| 模型 | 大小 | Top1准确率 | Top5准确率 | 参数数目 | 深度 |
| --- | --- | --- | --- | --- | --- |
| Xception | 88MB | 0.790 | 0.945 | 22,910,480 | 126 |
| VGG16 | 528MB | 0.715 | 0.901 | 138,357,544 | 23 |
| VGG19 | 549MB | 0.727 | 0.910 | 143,667,240 | 26 |
| ResNet50 | 99MB | 0.759 | 0.929 | 25,636,712 | 168 |
| InceptionV3 | 92MB | 0.788 | 0.944 | 23,851,784 | 159 |
| IncetionResNetV2 | 215MB | 0.804 | 0.953 | 55,873,736 | 572 |
| MobileNet | 17MB | 0.665 | 0.871 | 4,253,864 | 88 |

由表4-3 本次实验选择VGG16为本次特征提取的卷积神经网络预训练模型，原因是VGG16神经网络深度浅，适合没有GPU等高性能浮点运算设备使用。

特征提取模块流程图如图4-2所示，在进行特征提取时，对图像进行预处理，缩小图像的分辨率，降低卷积运算时的性能消耗，提升系统稳定性。特征提取处理之后，对所得特征值进行归一化，特征值提高运算速度，将特征值压缩到0-1之间。

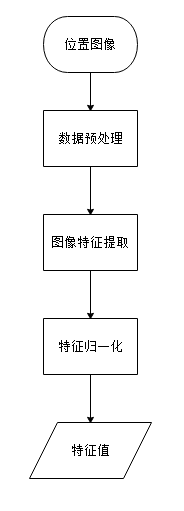


图4-2 特征提取流程图

特征提取模块算法描述如下表4-4所示，使用了VGG模型和OpenCV支持库，使用特征功能和归一化特征向量。

表4-4 特征提取伪代码

| 算法1：特征提取 |
| --- |
| BEGIN  定义图像大小  定义VGG16模型  使用OpenCV对原始图像进行变换成定义的图像大小  图像向量化  对图像向量进行处理，改变向量形状  对图像进行预处理  输出VGG16模型的预测值  对预测值进行归一化处理  返回特征值  END |

特征提取模块函数设计如表4-5所示：

表4-5 特征提取模块涉及到的常量和函数

| 常量 | input\_shape = (272, 480, 3) |
| --- | --- |
| 函数 | imageAdjust(image,width,height)  extract\_feat(img\_path) |

特征提取程序函数具体代码实现如下代码所示：

代码 imageAdjust函数实现与说明

def imageAdjust(self,image,width,height):

# 读取图像文件

\_img = imread (image)

# 图像变换

res = resize (\_img, (width, height), interpolation=INTER\_AREA)

# 创建存储器

img = Image.fromarray (cvtColor (res, COLOR\_BGR2RGB))

#返回变换后的图像

return img

代码 extract\_feat函数实现与说明

def extract\_feat(self,img\_path):

# 定义图像大小

input\_shape = (272, 480, 3)

# 定义Keras的预训练模型VGG16的参数

model = VGG16(input\_shape = (input\_shape[0],input\_shape[1],input\_shape[2]), pooling = 'max', include\_top = False)

# 调用图像变换函数

feat = feature()

img = feat.imageAdjust(img\_path,input\_shape[1],input\_shape[0])

# 图像向量化

img = image.img\_to\_array(img)

# 改变向量形状

img = np.expand\_dims(img, axis=0)

# 图像进行预处理

img = preprocess\_input(img)

# 使用VGG16进行预测

feat = model.predict(img)

# 特征值归一化

norm\_feat = feat[0]/LA.norm(feat[0])

# 返回特征值

return norm\_feat

## 特征数据库模块

特征数据库模块主要的功能为读取数据库特征值、写入特征值、插入特征值。特征数据库的数据结构如图4-3所示。每个特征值对应一个索引，同时每个图像名称对应一个索引，这两种数据类型的索引中的ID是相等的。如 [特征索引1] 对应 [名称索引1]，这样可以根据特征索引得到相应的图像名称索引，从而得出图像名称。

数据库数据结构设计如下图所示：

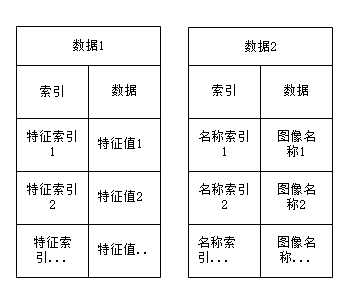


图4-3 数据库数据结构设计

特征数据库设计流程图如图4-4所示，特征数据库操作有三种，插入操作、写操作、读操作。

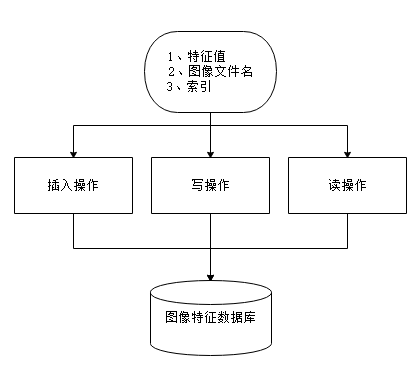


图4-4 特征数据库流程图

特征数据库算法描述，如表4-6和表4-7所示：

表4-6 特征数据库写入功能伪代码

| 算法2：特征数据库写入 |
| --- |
| BEGIN  定义名称列表  对名称列表进行编码  以追加类型打开h5文件  写入特征值和特征值索引  写入图像名称和名称索引  END |

表4-7 特征数据库读取功能伪代码

| 算法3：特征数据库读取 |
| --- |
| BEGIN  以读类型打开h5文件  按指定特征索引读取特征值列表  按指定名称索引读取图像名称列表  对名称列表进行解码  返回目标数据  END |

特征数据库模块函数如表4-8所示：

表4-8 特征数据库模块涉及到的函数

| 函数 | wH5FileData(Key,feats,names,filename)  rH5FileData(Key,filename) |
| --- | --- |

特征数据库代码实现以及说明如下代码所示：

代码 wH5FileData函数实现与说明

def wH5FileData(Key,feats,names,filename):

# 名称列表

namess = []

# 对名称列表进行编码

namess = strEncode(names)

try:

# 以追加类型打开h5文件

with h5py.File (filename, 'a') as h5f:

# 写入特征值和特征值索引

h5f.create\_dataset ("data"+str(Key), data=feats)

# 写入图像名称和名称索

h5f.create\_dataset ("name"+str(Key), data=namess)

# 写入失败则异常

except RuntimeError:

raise NameError('Unable to create link (name already exists)')

return 0

代码 rH5FileData函数实现与说明

def rH5FileData(Key,filename):

try:

# 以读类型打开h5文件

with h5py.File (filename, 'r') as h5f:

# 按指定特征索引读取特征值列表

feats = h5f["data" + str (Key)][:]

# 按指定名称索引读取图像名称列表

imgNames = h5f["name" + str (Key)][:]

# 返回特征值和解码后的名称

return feats,imgNames[0].decode("utf-8")

# 读取失败则异常

except KeyError:

print("Read HDF5 File Key Error")

return 1

## 特征检索模块

特征检索模块首先需要调用特征提取模块对用户提供的图像进行特征提取，然后读取特征数据库中的所有数据，最后进行匹配及其评分，输出检索结果。

特征检索流程图如图4-5所示：

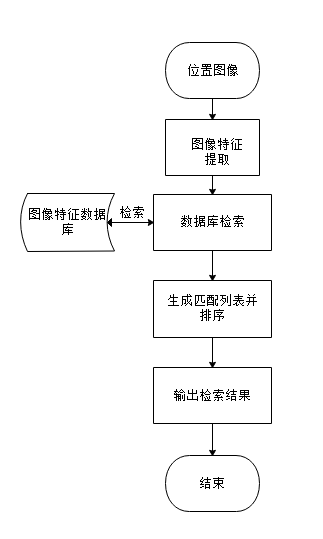


图4-5 特征检索流程图

特征数据库算法描述，如表4-9所示：

表4-9 特征检索伪代码

| 算法2：特征检索评分 |
| --- |
| BEGIN  对输出图像进行特征提取  将得到的特征和数据库读取的特征进行向量点积  将点积后的值从大到小排序  对排序完成的向量进行评分  返回评分列表  END |

特征检索模块所需函数，如表4-10所示：

表4-10 特征检索模块涉及到的函数

| 函数 | featureSearch(queryImage,feats)  showImage(queryImage,imlist,result) |
| --- | --- |

特征检索模块，具体代码实现，核心代码如下代码所示：

代码 featureSearch函数实现与说明

def featureSearch(queryImage,feats):

feat = feature()

# 特征提取

queryVec = feat.extract\_feat (queryImage)

# 向量点积

scores = np.dot (queryVec, feats.T)

# 点积后排序

rank\_ID = np.argsort (scores)[::-1]

# 根据排序ID对其评分

rank\_score = scores[rank\_ID]

# 返回排序ID和评分

return rank\_ID,rank\_score

根据评分ID可以获取图像名称，最后利用图像名称查得图像位置信息。最后返回位置信息和显示最相似图像。

代码 showImage函数实现与说明

def showImage(queryImage, imlist, result):

# 读取和显示查询图像

\_queryImg = mpimg.imread (queryImage)

plt.title ("Query Image")

plt.imshow (\_queryImg)

plt.show ()

# 显示检索结果

for i, im in enumerate (imlist):

image = mpimg.imread (result + "/" + im)

plt.title ("top %d" % (i + 1))

plt.imshow (image)

plt.show ()

return 0

# 系统测试与分析

本章介绍了功能测试和性能测试的测试方法和测试用例设计，测试用例的执行结果。

## 实验数据集

图像数据集为人工采集的北京邮电大学校内图像，一共有8062张图像文件，还有图像与之对应的8062个图像指纹信息文本文件。共计162个位置信息，图像内容为，从多个角度对同一位置目标进行拍摄。

图像有1088x1920/1152x2048/1232x2048/1440x2560/1520x2688等5种图像分辨率。

图像数据集有两部分：训练数据集、测试数据集。

训练数据集：6，442张图像，总大小为4.32GB。每个位置的图像有38-40张。

测试数据集：1，620张图像，总大小为1.07GB，每个位置的图像有10张。

## 测试环境

本次系统测试采用编写测试用例函数，对相对应的功能和性能进行测试。

本次系统测试检查以下问题：

1、图像特征提取功能是否正常

2、图像特征数据库功能是否正常

3、图像特征检索功能是否正常

4、图像特征提取耗时是否达标

5、图像检索耗时是否达标

6、图像检索准确率是否达标

7、图像检索召回率是否合理

本次系统测试的测试环境如表5-1所示：

表5-1 测试环境

| 测试环境 | 硬件 | 处理器：Inter i5-6200U  显卡：Inter HD 520  内存容量：12GB |
| --- | --- | --- |
| 软件 | 操作系统：Windows 10 企业版 64bit  IDE：PyCharm 2018.1 64bit |
| 设备 | Thinkpad S2笔记本 |

测试用例表设计如表5-2所示，主要测试三个功能和三个特征检索系统性能，三个功能为特征提取、特征数据库、特征检索，三个系统性能为检索耗时、检索准确率、检索召回率。

表5-2 测试用例表

| 测试用例 | 测试目标 | 测试结果 | 存在问题 | 优化方向 |
| --- | --- | --- | --- | --- |
| 特征提取功能 | 是否能提取图像特征 | 待测 | 待测 | 待测 |
| 特征数据库功能 | 是否能正常读写特征数据库 | 待测 | 待测 | 待测 |
| 特征检索功能 | 是否能成功检索出最相似图像，且返回位置信息 | 待测 | 待测 | 待测 |
| 特征提取耗时 | 特征提取性能耗时是否合理，是否达到预期目标 | 待测 | 待测 | 待测 |
| 检索耗时 | 系统检索耗时是否达到预期目标 | 待测 | 待测 | 待测 |
| 检索准确率 | 准确率是否到达预期目标 | 待测 | 待测 | 待测 |
| 检索召回率 | 召回率高，则准确率第；准确率高，则召回率低 | 待测 | 待测 | 待测 |

## 功能测试

本次功能测试方法为，测试三个主要的功能模块，编写测试用例进行测试，下面详细介绍三个功能模块的测试用例。

### 图像特征提取功能

图像特征提取功能模块测试用例设计为，输入一张图像，得到图像特征值和特征值类型和形状。核心代码如代码所示：

代码 特征提取测试代码

def testExtractFeat(img):

f = feature ()

t = time()

norm\_feat = f.extract\_feat (img)

print ("特征提取耗时(秒)：%.2f s" % (time () - t))

print ("特征向量形状: ", norm\_feat.shape)

print ("特征向量大小: ", norm\_feat.size)

print ("特征向量类型: ", type (norm\_feat))

return 0



图5-1 特征提取测试图像

输入测试图像，测试图像如图5-1所示，测试图分辨率为1088x1920，大小为245KB，测试结果如下图5-2所示：

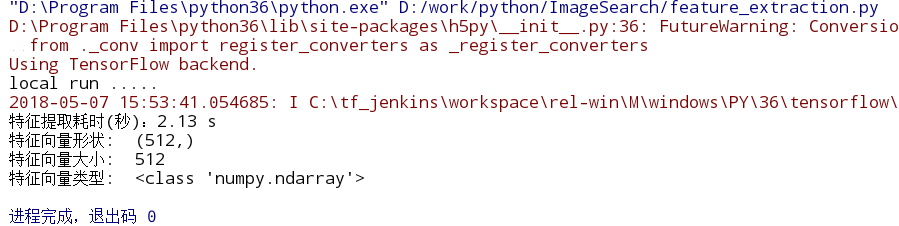


图5-2 特征提取测试结果截图

由结果得知，特性特征提取功能内正确提取特征值，特征值形状为(512,)，特征大小为512，特征值类型为numpy.ndarray。

### 图像特征数据库功能

图像特征数据库功能模块测试用例设计为，使用图像特征提取功能提取出特定图像的特征值，然后将图像特征值和图像名称存入文件，最后取出图像特征值和图像名称，验证图像特征数据库功能模块是否能成功读写图像特征数据库，是否能成功将图像特征插入图像特征数据库。

核心代码如下所示：

代码 特征数据库测试代码

def testDatabase():

featsList = []

nameList = []

h5filename = "./models/image\_Feature\_Test.h5"

dataset = "./imagetest"

# 取图像数据集列表

img\_list = b.getFileList (dataset, "JPEG")

# 提取图像数据集列表特征后存入HDF5文件

t = time()

etlFeature (showHDF5Len (h5filename), img\_list, h5filename)

print ("特征数据库写入耗时(秒)：%.2f s" % (time () - t))

# 读取图像特征值和图像名称

t = time ()

for i in range (showHDF5Len (h5filename)):

feats, imgNames = rH5FileData (i, h5filename)

featsList.append (feats)

nameList.append (imgNames)

print ("特征数据库读取耗时(秒)：%.2f s" % (time () - t))

print("特征值列表长度：", len(featsList))

print("特征值对应的图像名称列表：", nameList)

print("图像名称列表长度:", len(nameList))

return 0

使用三张测试图像作为特征数据库测试图像，测试图像如下：



图5-3 特征数据库测试图像一

图5-3的位置信息为“校车车库”，作为特征数据库测试图像一。分辨率为1088X1920，图像大小为214KB。



图5-4 特征数据库测试图像二

图5-4的位置信息为“联通报刊亭”，作为特征数据库测试图像二。分辨率为1088X1920，图像大小为238KB。



图5-5 特征数据库测试图像三

图5-5的位置信息为“联通报刊亭”，作为特征数据库测试图像三。分辨率为1088X1920，图像大小为192KB。

三张测试图像为随机挑选的测试图像，具有测试原则的随机性原则，将以上三张测试图像输入，进行特征值提取，最后存入特征数据库文件。观察特征数据库是否能进行特征存储和特征数据库是否具有特征值插入功能。观察特征数据库的调整数据存储文件大小，大小是否符合要求。特征数据库文件大小会过大影响特征数据库的读写性能，需要把特征数据库文件大小控制在一定的范围内。



图5-6 特征数据库测试结果截图

如测试结果图5-6可知，输入的三张图的写入时长为6.49秒，其中耗时包括特征提取耗时。

再次执行该测试用例则将重复的三张图像的特征值插入图像特征数据库，如图5-7所示，该图像特征数据库有特征值插入的功能。

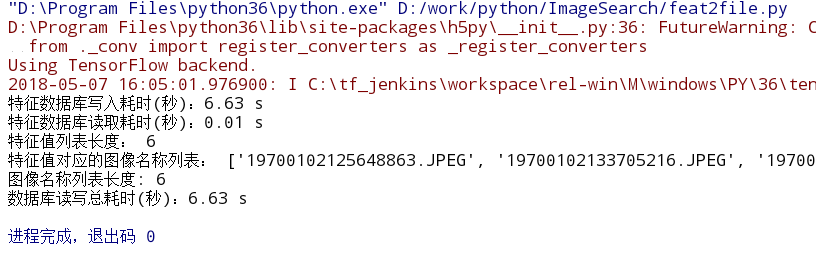


图5-7 特征数据库测试结果截图

由测试结果图5-7可知，写入特征成功，且成功读取特征值，特征值数据库功能模块测试通过。

本系统对其训练集进行特征提取和特征存储，训练集特征数据库文件大小为12.7MB，大小合理，节约系统空间，为大数据量提供基础。保证以较小的空间存储较为合理的特征值。

### 图像特征检索功能

图像特征检索功能模块测试用例设计为，指定一个已存在的图像特征数据库和一张图像测试集中的测试图像，验证特征检索功能模块是否能成功检索出最相似的图像，并且显示相似度最高的3张图像。

核心代码如下所示：

代码 特征检索测试代码

def SearchTest (resultnum,queryImage,ModelFile,imageinfopath):

start2 = time.time ()

outputInfo = ""

feats, imgNames = rH5FileData2 ("feature", "imagename", ModelFile)

rank\_ID, rank\_score = featureSearch(queryImage,feats)

imList, scoresList = getSearchResult(resultnum,imgNames,rank\_ID,rank\_score)

imgInfoList = getImageInfo(imList,imageinfopath)

\_imageInfo = getImageInfo (queryImage, imageinfopath) # 获取图片信息

if \_imageInfo == imgInfoList[0]:

outputInfo = "完美匹配！"

elif \_imageInfo == imgInfoList[1]:

outputInfo = "次要匹配！"

else:

outputInfo = "匹配失败！"

print ("原图为 :", queryImage)

print ("原图信息:",\_imageInfo)

print ("最高相似度的%d张图片为: " % resultnum, imList)

print ("最高%d张图片的评分：" % resultnum, scoresList)

print ("图片信息为: ", imgInfoList)

print ("搜索结果 ：",outputInfo)

print ("本次检索总耗时(秒)：%.2f s" % (time.time () - start2))

# 显示图像

showimage(queryImage,imList,result)

return 0

输入测试图像，如图5-8，测试图像的位置信息为“北邮网络中心”，分辨率为1088X1920，图像大小为264.5 KB



图5-8 特征检索测试图像

测试结果，如图5-9所示:

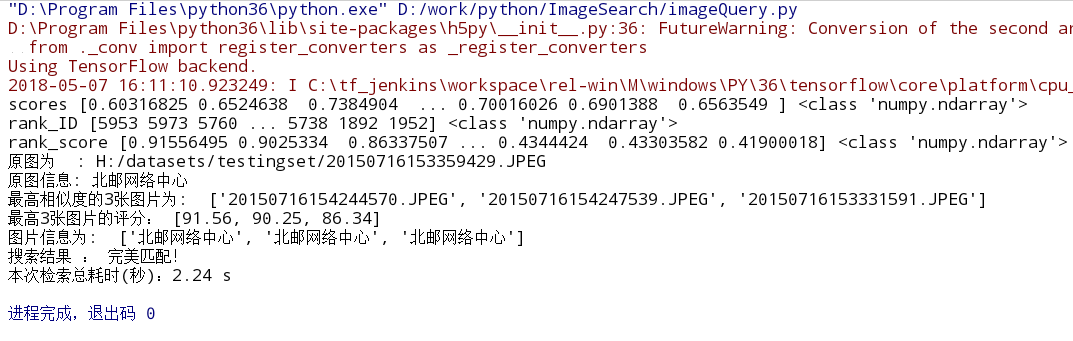


图5-9 特征检索测试结果截图

特征检索测试结果图5-9，特征检索测试结果能正确匹配出测试图像的位置信息，检索出三张最相似图像，分别为top1如图5-10、top2如图5-11、top3如图5-12，匹配分数分别为91.56、90.25、86.34，匹配程度高，top1-3匹配分数都在80分以上。



图5-10 特征检索测试结果图像TOP1

如图5-10所示，top1的图像为最相似图像，相似度分数为91.56分，信息为“北邮网络中心”，由此可知，本次特征检索测试通过，能检索出最相似图像，且返回图像位置信息。



图5-11 特征检索测试结果图像TOP2

如图5-11，top2的图像也为相似图像，相似度分数为90.25分，信息为 “北邮网络中心”。由此可知，top2也符合位置信息。



图5-12 特征检索测试结果图像TOP3

如图5-12，top3的图像也为相似图像，相似度分数为86.34分，信息为 “北邮网络中心”。由此可知，top3也符合位置信息。

由以上测试结果得知，该测试用例成功获取图像检索结果，并且得到相似度最高的三张图像TOP1-3。位置信息正确匹配，检索图像的位置信息为“北邮网络中心”，检索结果为“北邮网络中心”，成功匹配位置信息。

## 性能测试

### 特征提取测试

特征提取主要性能耗时为VGG16预测的过程耗时，卷积神经网络深度越深耗时越长，设备浮点计算性能越好预测耗时越短。

本次特征提取耗时性能测试，测试用例设计为，对随机的图像进行多次特征提取，计算其平均性能耗时。

测试用例代码如下所示：

def proTest():

img\_path = "./imagetest/19700102134147686.JPEG"

t = time ()

for \_ in range(100):

f = feature ()

norm\_feat = f.extract\_feat (img\_path)

print ("特征提取耗时(秒)：%.2f s" % ((time () - t)/100))

return 0

测试结果为2.16秒，则特征提取平均耗时为2.16 秒。

### 特征检索测试

特征检索性能测试的方法为，重复对部分特殊的图像特征进行测试，测试取其时间耗时平均值。

本次测试用例设计为，从测试集中选取10张图像，每种分辨率两张，按照图像大小选取两张图片，最大的和最小的。每次查询测试测试10次，计算消耗时间的平均值，制作成表格和柱状图。

经过测试用例得到性能测试的测试结果为：每次平均检索耗时为2.289秒。

表 5-3 检索性能测试表

|  | 1088 x 1920 | 1152 x 2048 | 1232 x 2048 | 1440 x 2560 | 1520 x 2688 |
| --- | --- | --- | --- | --- | --- |
| 527Kb | 2.29 |  |  |  |  |
| 130Kb | 2.23 |  |  |  |  |
| 1.53Mb |  | 2.31 |  |  |  |
| 380Kb |  | 2.24 |  |  |  |
| 1.79Mb |  |  | 2.32 |  |  |
| 739Kb |  |  | 2.28 |  |  |
| 1.13Mb |  |  |  | 2.33 |  |
| 283Kb |  |  |  | 2.27 |  |
| 1.14Mb |  |  |  |  | 2.35 |
| 440Kb |  |  |  |  | 2.27 |
| 平均值 | 2.26 | 2.275 | 2.30 | 2.30 | 2.31 |

将性能测试表5-3转化为柱状图，如图5-13所示：

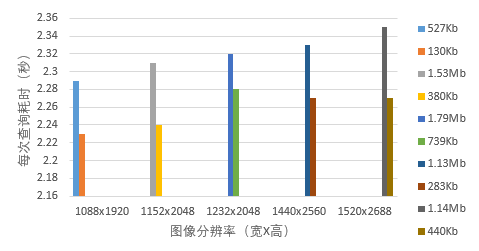


图5-13 系统测试性能柱状图

### 特征检索准确率测试

准确率是对检索结果的一种评价，准确率表示“预测正确的样本”与“总样本”的比例。能解释有多大概率能检测到正确的样本，准确率为检索系统的重要评价指标和常用评价指标。

准确率测试方法为，使用图像测试集对图像训练集生成的图像特征值数据库进行验证，编写测试用例程序，计算特征检索的准确率。

测试用例设计为，将1620张测试集图像分别进行检索。由于图像集数量过大，笔记本负载压力大，容易造成程序不稳定，系统崩溃，需要进行分批进行测试。200张图像为一批，一共分为8个批次。特征检索准确率测试代码如下所示：

代码 性能测试代码

def testSetTest(testSet,imageinfopath,feats,imgNames):

resultnum = 2

imageList = getImageList (testSet)

probability = 00.00

num = len (imageList)

errorNum = 0

for i in imageList:

rank\_ID, rank\_score = featureSearch (i, feats) # 获取图像得分信息

imList, scoresList = getSearchResult (resultnum, imgNames, rank\_ID, rank\_score)

\_imageInfo1 = getImageInfo (i, imageinfopath) # 获取图片信息

\_imageInfo2 = getImageInfo (imList[0], imageinfopath)

if \_imageInfo1 != \_imageInfo2:

errorNum += 1

print ("测试集错误的图片信息： ", \_imageInfo1)

print ("错误的图片最匹配的图片信息：", \_imageInfo2)

print ("测试集错误的图片： ", i)

print ("错误的图片最匹配的图片：", imList[0])

print ("相似度评分：", scoresList)

probability = errorNum / num

print ("错误率: %.2f%%" % (probability\*100))

return 0

测试结果如表5-4所示：

表5-4 准确率测试结果

|  | 错误匹配数（张） | 总数（张） | 正确匹配率（%） |
| --- | --- | --- | --- |
| 第1批次测试 | 10 | 200 | 95 |
| 第2批次测试 | 2 | 200 | 99 |
| 第3批次测试 | 8 | 200 | 96 |
| 第4批次测试 | 8 | 200 | 96 |
| 第5批次测试 | 13 | 200 | 93.5 |
| 第6批次测试 | 13 | 200 | 93.5 |
| 第7批次测试 | 10 | 200 | 95 |
| 第8批次测试 | 20 | 220 | 90.91 |

根据本次测试用例的测试结果得出，图像检索的正确率约为94.81%。

### 特征检索召回率测试

召回率为被“检测到的样本”和“应该被检测到的样本”的比例，准确率和召回率都有助于系统检索性能相关性的理解，召回率和准确率是检索系统的主要评价指标。一般而言，召回率高，则准确率就低，准确率高，则召回率低。召回率测试方法为，使用图像测试集对图像训练集生成的图像特征值数据库进行验证，判断特征检索的召回率。

测试用例设计为，将1620张测试集图像分别进行检索。由于图像数据集数量过大，进行随机选取部分测试集图像进行召回率测试。

本次召回率测试用例代码如下：

代码 召回率测试

def recallRate():

recallList = []

\_testImgList = b.getFileList("H:/datasets/testingset","JPEG")

import random

testList = random.sample(\_testImgList,100)

feats, imgNames = rH5FileData2 ("feature", "imagename", "./models/imageCNNAll.h5")

for queryImage in testList:

rank\_ID, rank\_score = featureSearch (queryImage, feats)

imList, scoresList = getSearchResult (100, imgNames, rank\_ID, rank\_score)

imgInfoList = getImageInfo (imList, "H:/datasets/imageinfo")

\_imageInfo = getImageInfo (queryImage, "H:/datasets/imageinfo")

count = imgInfoList.count(\_imageInfo)

recallList.append(count/40)

print(recallList)

print("recall rate:",(sum(recallList)/len(recallList)))

return 0

根据召回率公式，设计测试用例。将输出显示图像设置为40张，判断输出图像中有多少张图像是符合要求的，然后处于相似图像总数，得到一次测试的数据，最后进行多次测试，统计实验结果。

测试结果如图5-14所示：

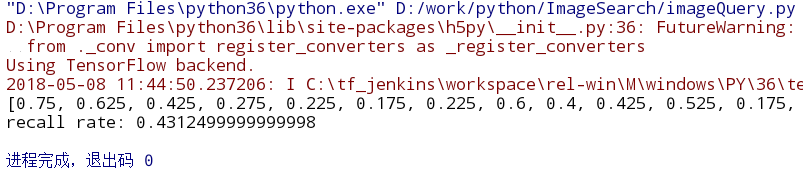


图5-14 召回率测试结果

根据图5-14所知，通过多次随机获取的图像进行测试，本次测试结果所得召回率为43.125%。

## 测试结果分析

本次测试总共测试了功能模块和系统性能，功能模块分别测试了图像特征提取功能、图像特征数据库功能、特征检索功能等三个主要功能。系统性能分别测试了特征提取性能、特征检索性能、准确率、召回率等四个性能评价指标。其中，准确率和召回率为主要评价指标，其余为次要评价指标。

根据系统功能测试和系统性能测试所得数据和测试用例设计表，可以汇总得到表5-5，使其更为直观的看到系统的总体性能。

本次功能测试特征提取功能测试通过，能成功提取特征值。特征数据库功能测试通过，能对特征向量进行写入读取等操作，还可以进行特征值插入操作。特征检索功能测试通过，能在指定时间内检索出位置信息和最相似图像。

从测试用例表5-5得出，本章节系统测试通过，测试测试结果符合原定测试目标。但是某些功能模块还存在着某些不足，如特征数据库的删除功能还没完善，暂定作为下一步优化。

从系统检索性能测试结果绘画出的柱状图可得到，系统的查询耗时与图像的大小和分辨率有关。分辨率越大检索耗时越长，分辨率与检索耗时成正相关。图像大小越大检索时长越大，图像大小与检索耗时成正相关。

系统性能瓶颈的主要性能消耗在提取图像特征值的过程，也就是卷积神经网络的运算过程。设备性能和神经网络层数影响特征提取过程。

系统测试结果汇总，如表5-5所示：

表5-5 测试用例表

| 测试用例 | 测试目标 | 测试结果 | 存在问题 | 优化方向 |
| --- | --- | --- | --- | --- |
| 特征提取功能 | 是否能提取图像特征 | 测试通过 | 无 | 加深卷积神经网络深度，提升特征提取性能 |
| 特征数据库功能 | 是否能正常读写 | 测试通过 | 暂无删除功能 | 增加删除功能或者使用相应数据库存储 |
| 特征检索功能 | 是否能成检索出所需图像 | 测试通过 | 无 | 支持web URL的使用 |
| 特征提取性能 | 特征提取性能耗时是否合理，是否达到预期目标 | 提取耗时2.16秒 | 无 | 提升设备性能或者优化神经网络层数 |
| 检索性能 | 系统检索耗时是否小于3秒 | 检索耗时2.29秒 | 无 | 依赖于特征提取的性能 |
| 系统检索准确率 | 准确率需要在90%以上 | 准确率94.81% | 无 | 加深卷积神经网络深度，提升特征提取性能 |
| 系统检索召回率 | 召回率高，则准确率第；准确率高，则召回率低 | 召回率43.125% | 无 | 本次系统不需要对召回率进行优化 |

由表5-5可知，特征提取功能需要下一步进行优化。特征提取功能提升准确率，需要加深卷积神经网络的深度，除了加深网络之外还可以选择较优的函数，使其快速收敛，提升特征提取性能，缩短特征提取时间和特征提取准确率。

由表5-5可知，对大型的图像库进行检索，特征数据库功能需要使用相应的数据库进行存储，如Mongodb、Redis等NoSQL数据库，能够对数据进行快速读写和基本的增删改查功能。如果图像数据库太大，可以考虑使用分布式。

由表5-5可知，检索功能还有很大的提升空间。特征检索目前使用命令行方式使用，优化方向为支持浏览器的web页面进行检索，或者移动端APP进行检索。

由表5-5可知，检索所消耗的时间虽然在合理范围内，但是用户体验达不到要求。需要进一步提升，把检索时间控制在一秒内，提升用户交互。

本次性能测试结果中准确率和召回率符合系统，本系统准确率高，召回率则低。两者达到一个平衡点，所以系统性能测试的真实性和可靠性得到保证。最理想的情况下是准确率与召回率两者同时为高，但是由于许多客观因素存在，导致出现某种平衡。准确率保证检索系统任务检索的确定性，而召回率度量可用于疾病监控、反垃圾邮件等等方面的衡量。

本次性能测试还有许多不足之处，性能测试的设备可使用专业浮点计算设备。在专业情况下，本系统的各方面的表现如何，由于本次测试条件限制，暂无专业设备。后续改进中可使用Google的TPU或者GPU进行系统性能测试，观察设备浮点计算能力是否对本系统具有较大影响。

结束语

本文主要介绍了基于深度学习的图像搜索技术融合目标位置识别的方法，做了相关实验验证了该方法的可行性与性能。首先通过分析介绍了相关实验背景和研究现状确定了实验的需求及其主要实验的实现过程。本文课题提出一种高准确性、高性能、快速响应的目标位置识别方法，改善目前定位技术的精度不高且仅仅能够获取用户的物理位置信息，而不能识别出用户所感兴趣的目标位置的逻辑含义的缺陷。然后，对课题的实验过程中涉及的卷积神经网络、基于深度学习的图像检索技术以及基于目标定位技术等相关信息进行了详细探讨与总结。本文最终设计并实现了基于深度学习的位置识别系统。为了对该系统进行功能和性能上的测试，在北京邮电大学校园内采集了大量目标的位置信息构建用来训练和测试的数据集，并对系统的功能和性能两方面进行了测试分析。通过对系统的测试，验证了基于深度学习的位置识别方法的正确性及系统的可用性。

完成了该系统的设计与实现对我有很大的提升，提高了我的实际操作能力和系统架构能力。在此同时也认识到了自己的不足，在开发过程中，遇到很多问题，同时也解决了很多问题。

该系统仍然存在着许多不足之处，存在以下几点可以改进：

本系统使用的方法中的图像数据库需要依赖于人工采集和维护，在后续的工作中，可以采用图像采集车进行采集，也可以使用用户上传的数据，动态更新图像数据库。

本文提出的方法进行特征提取和识别的过程在服务器端进行，后续工作中，希望通过进一步研究特征提取和识别的复杂度，降低特征维度，提升识别的准确性，降低识别所需时间，然后应用于移动端，提升系统的实用性。

本文提出的方法仅从图像的角度，距离等特征信息方面考虑，在后续的工作中，可以进一步考虑时间、季节、天气、图像遮挡、图像模糊等因素，提高系统的可用性和准确率。

本文提出的位置识别系统底层依赖于卷积神经网络VGG16预训练模型，该模型存在许多不足，后期为提升性能选择其他模型，增加其神经网络深度，提升图像检索准确率。

参考文献

1. 麦好.机器学习实践指南[M].机械工业出版社,2016.
2. 刘凡平.大数据时代的算法：机器学习、人工智能及其典型实例[M].电子工业出版社,2017.
3. 库姆斯 (Alexander T.Combs).Python机器学习实践指南[M].人民邮电出版社,2017.
4. 李嘉璇.TensorFlow技术解析与实战[M].人民邮电出版社,2017.
5. Sebastian Raschka.python机器学习[M].机械工业出版社,2017.
6. 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.
7. 何宇健.Python与机器学习实践[M].电子工业出版社,2017.
8. Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer Science, 2014.
9. Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[C]// International Conference on Neural Information Processing Systems. Curran Associates Inc. 2012:1097-1105.
10. Sermanet P, Eigen D, Zhang X, et al. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks[J]. Eprint Arxiv, 2013.
11. Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the Inception Architecture for Computer Vision[J]. Computer Science, 2015:2818-2826.
12. Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[J]. 2014:1-9.
13. Alom M Z, Taha T M, Yakopcic C, et al. The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches[J]. 2018.
14. 刘成. LBS 定位技术研究与发展现状 [J]. 导航定位学报, 2013 (1): 78-83.
15. Schindler G, Brown M, Szeliski R. City-scale location recognition [C]// IEEE Conference on Computer Vision and Pattern Recognition, 2007: 1-7.
16. Chen D M, Baatz G, Koser K, et al. City-scale landmark identification on mobile devices[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2011:737-744.
17. Guan T, He Y, Gao J, et al. On-Device Mobile Visual Location Recognition by Integrating Vision and Inertial Sensors[J]. IEEE Transactions on Multimedia, 2013, 15(7):1688-1699.
18. Guan T, He Y, Duan L, et al. Efficient BOF Generation and Compression for On-Device Mobile Visual Location Recognition[J]. IEEE Multimedia, 2014, 21(2):32-41.
19. 王丽君,于莲芝.基于卷积神经网络的位置识别[J].电子科技,2017,30(01):104-106+114.
20. Paek, Jeongyeup, Kim, et al. Energy-efficient rate-adaptive GPS-based positioning for smartphones[J]. Proceedings of Acm Mobisys ’, 2010:299-314.
21. 李海燕,张岩.移动通信网络的移动台定位技术及应用[J].邮电设计技术,2006(03):27-34.
22. 徐怀宇. 基于手机多传感器融合的室外目标位置识别方法研究[D].北京邮电大学,2017.

附录

致谢

时间易逝，岁月如光。此次论文的撰写工作即将完成，同时也预示我四年的大学学习生涯逐渐进入尾声。回顾这四年的在校学习的美好时光，遗憾与满足并存。虽然有遗憾，但是更多的是开心与满足。我为此生能有如此一段特殊的经历感到开心，对在此期间遇到众多关心帮助我的人表达最真诚的敬意和感谢。

首先，感谢我的导师王浩老师。老师专业学识渊博，教学态度严谨，在论文与毕业设计撰写期间，王浩老师细心指导、教会了我如何设计严谨的实验过程和记录实验结果。此次论文和毕业设计撰写过程，我受益匪浅。在论文完成过程中，王浩老师解答了我心中的许多疑问，提供了许多宝贵的意见。在此，向王浩老师表达我最诚挚的敬意和衷心的感谢。

其次，感谢我的舍友们，我们一起上课，一起学习，一起旅游，与你们的情谊是我宝贵的财富，感谢你们的包容和关心，在你们四年的陪伴下，我度过了一个令我难忘的大学生活。

另外，感谢百度学术和谷歌学术两个搜索引擎，帮助我阅读更专业的论文，提升我的专业水平。让我在论文和毕业设计撰写过程中避免了很多错误，希望你们越做越好。希望你们能继续给学术界提供专业的学术论文搜索服务。

最后，感谢我的父母和女朋友。你们在生活中给予我无微不至的关怀，在我绝望的时候激励我，在我有所收获的时候为我感到开心，鼓励我再接再厉。你们是我努力的动力，感谢你们一直以来给我最无私的关怀和陪伴，谢谢你们。