基于深度学习的图像检索系统①

胡二雷[1], 冯 瑞[2]

[1](复旦大学 计算机科学技术学院, 上海 201203)
[2](上海市智能信息处理重点实验室 上海视频技术与系统工程研究中心, 上海 201203)

摘 要: 基于内容的图像检索系统关键的技术是有效图像特征的获取和相似度匹配策略. 在过去, 基于内容的图像检索系统主要使用低级的可视化特征, 无法得到满意的检索结果, 所以尽管在基于内容的图像检索上花费了很大的努力, 但是基于内容的图像检索依旧是计算机视觉领域中的一个挑战. 在基于内容的图像检索系统中, 存在的最大的问题是“语义鸿沟”, 即机器从低级的可视化特征得到的相似性和人从高级的语义特征得到的相似性之间的不同. 传统的基于内容的图像检索系统, 只是在低级的可视化特征上学习图像的特征, 无法有效的解决“语义鸿沟”. 近些年, 深度学习技术的快速发展给我们提供了希望. 深度学习源于人工神经网络的研究, 深度学习通过组合低级的特征形成更加抽象的高层表示属性类别或者特征, 以发现数据的分布规律, 这是其他算法无法实现的. 受深度学习在计算机视觉、语音识别、自然语言处理、图像与视频分析、多媒体等诸多领域取得巨大成功的启发, 本文将深度学习技术用于基于内容的图像检索, 以解决基于内容的图像检索系统中的“语义鸿沟”问题.

关键词: 基于内容的图像检索; 深度学习; 特征提取; 匹配

Image Retrieval System Based on Deep Learning

HU Er-Lei[1], FENG Rui[2]

[1](School of Computer Science, Fudan University, Shanghai 201203, China)
[2](Shanghai Key Laboratory of Intelligent Information Processing, Shanghai Engineering Research Center for Video Technology and System, Shanghai 201203, China)

Abstract: Learning effective feature representations and similarity measures are crucial to the retrieval performance of a content-based image retrieval system. In the past, the system works on the low-level visual features of input query image, which does not give satisfactory retrieval results, so, despite extensive research efforts for decades, it remains one of the most challenging problem in computer vision field. The main problem is the well-known “semantic gap”, which exists between low-level image pixels captured by machines and high-level semantic concepts perceived by human. In the past, the content-based image retrieval system only works on the low-level visual features, which cannot solve “semantic gap” issue. Recently, the fast development of deep learning brings hope for the issue. Deep learning roots from the research of artificial neural network. In order to form more abstract high-level, deep learning combines low-level features, finds the regularities of distribution, which is different from other algorithm. Inspired by recent successes of deep learning techniques for computer vision, speech recognition, natural language process, image and video analysis, multimedia, in this paper, we apply deep learning to solve the “semantic gap” issue in content-based image retrieval.

Key words: content-based image retrieval; deep learning; feature extracting; match

随着计算机技术和多媒体技术的快速发展, 大量的数字图像随之产生, 在海量的图像数据库中如何快速找到特定的图像就需要使用图像检索技术.

传统的基于内容的图像检索系统主要使用低级的

① 基金项目:国家科技支撑计划(2013BAH09F01);上海市科委科技创新行动计划(14511106900)
收稿时间:2016-07-10;收到修改稿时间:2016-09-20 [doi:10.15888/j.cnki.csa.005692]

视觉特征, 例如颜色、形状、纹理等, 使用的分类器大多是浅层分类器如 svm, 这些系统存在一个最大的问题是无法处理语义鸿沟[1]的问题(即机器从低级的可视化特征得到的相似性和人从高级的语义特征得到的相似性之间的不同). 所以, 尽管在图像检索这个问题上提出了一系列技术, 且取得了一定的成果, 但是由于语义鸿沟[1]的存在, 图像检索依旧是一个具有挑战性的难题, 从更高层次分析, 基于图像内容的检索属于人工智能领域的问题, 即有没有机器可以像人一样识别图像的内容. 在现阶段的所有的技术中, 机器学习技术是目前解决语言鸿沟这个问题最有前景的技术.

在机器学习中, 深度学习技术近些年得到了快速的发展, 是近十年来人工智能领域取得重要突破的技术. 深度学习技术在计算机视觉、语音识别、自然语言处理、图像与视频分析多媒体等方面取得了巨大的成功.

本文尝试着将深度学习用于图像检索, 以判断深度学习技术能否解决语义鸿沟以及解决的程度. 目前, 国内外使用深度学习技术处理基于内容的图像检索才刚刚起步, 还处于快速发展阶段.

在图像检索系统中, 往往匹配的时间消耗比较大. 对于这个问题, 文中, 我们将每幅图像对应的 n(实验中是 20 维)维特征向量映射到 n 个数据库表, 用这种方式来建立索引, 匹配的时候根据用例图像 n 维中最大值的下标 index 来确定图像的类别(softmax 分类器得到的 n 维特征向量的每一维代表属于这个类别的概率, n 维相加的结果为 1), 之后从对应的数据表 tables_i 中检索, 从实验结果来看, 利用这种方式可以避免扫描整个检索库, 从而可以成倍的提高检索的效率. 在图像检索系统中, 我们主要解决了下面 3 个问题:

(1) 如何训练一个好的模型, 将用例图像分类到正确的类别;

(2) 如何选择和建立索引, 即确定每个图像的特征向量;

(3) 如何选择距离匹配算法, 使得相似的图像之间的距离尽可能的小, 不同图像之间的距离尽可能的大.

为了解决上面的问题, 我们使用了开源的深度学习框架 Caffe[11], 实验中, 训练模型主要使用的数据集是 ImageNet[5], 训练了 20 个类别, 主要是巴士、摆钟、包菜、杯子、菠萝、菜花、草垛、草莓、茶壶、橙子、电视、独轮车、帆船、钢琴、海岸、红酒、无花果、雪山、汽车、珊瑚.

1 相关的工作

我们的工作涉及到深度学习、深度学习框架 Caffe、基于内容的图像检索、距离度量学习, 在这一节, 主要是介绍这几个工作.

1.1 深度学习

深度学习[2]是机器学习的一个分支, 深度学习的概念源于人工神经网络的研究. 含有隐层的多层感知器就是一种深度学习结构. 深度学习通过组合低级特征形成更加抽象的高层表示属性类别或者特征, 以发现数据的分布特征. 深度学习是多个研究领域交差的产物, 包括神经网络、图形化建模、优化、模式识别和信号处理等.

深度学习之所以能在近些年如此快速的发展起来, 主要得益于下面两个原因:

(1) 计算机硬件的迅速发展提供了强大的计算能力, 使得训练大规模的神经网络成为可能, 如高性能的 gpu 可以集成上千个核.

(2) 海量的被标记的数据的应用缓解了训练过拟合的问题, 在深度学习中, 数据是“引擎”, Imagenet 有上百万的标注数据.

实验中我们使用的 Alexnet[4]是卷积神经网络(CNN), 它是诸多深度学习技术的一种. Alexnet 神经网络有 8 层, 前 5 层为卷积层, 后 3 层为全连接层, 其中在第 1、2、5 卷积层的后面是 pooling 层. 卷积层通过参数共享来减少神经网络层之间的参数数量. Pooling 层子采样卷积层的输出层, 减少下一层输入数据的大小, 一般采用 2*2 的窗口来做 maxpooling 或者 avgpooling 运算, 这样可以减少 75%的数据.

1.2 深度学习框架 Caffe

深度学习框架 Caffe 是按照 Alexnet 模型设计的, Alexnet 在 2012 年 ImageNet 图像分类比赛中取得了第一名, 它是一种深度卷积神经网络(CNN).

Caffe 是 Alexnet 的具体实现, Caffe 用 C++语言编写, 运算速度快, 模型化好, 有开源社区支持, 在学术界和工业界有大量的用户. Caffe 一共有 8 层神经网络, 前面的 5 层是卷积层, 后面 3 层是全连接层, 网络结构如图 1 所示.

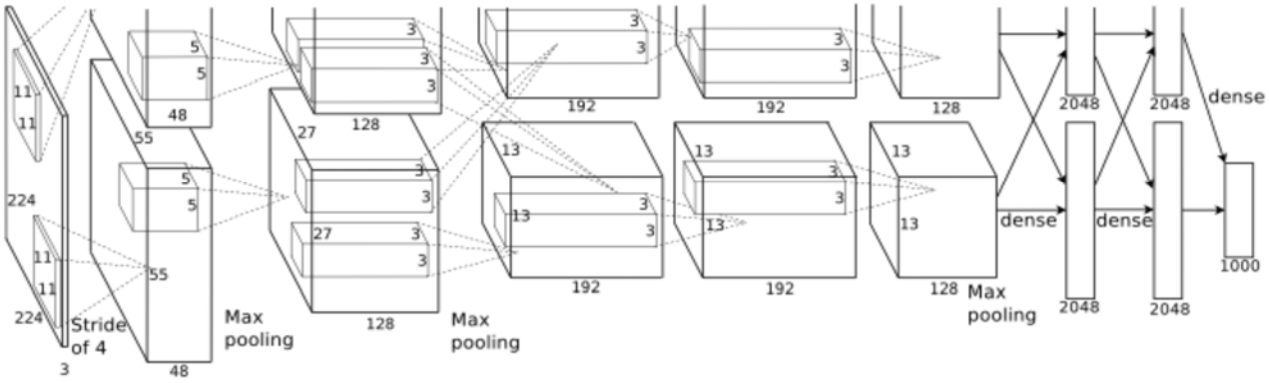

图 1 Caffe 网络结构

从图 1 可以看出，Caffe 的第 1、2、5 卷积层后面是 pooling 层，最后一层是 softmax 层，也是输出层.

图 2 到图 6 分别对应 Caffe 的前两层和后三层.

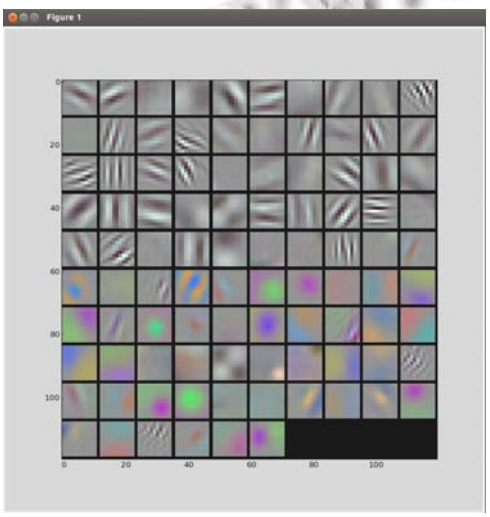

图 2 第一个卷积层网络结构

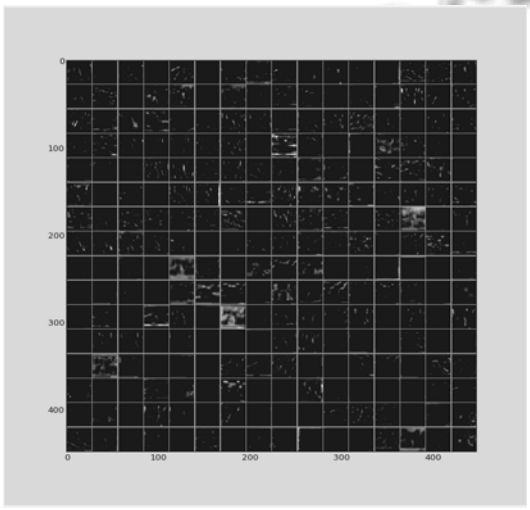

图 3 第二个卷积层网络结构

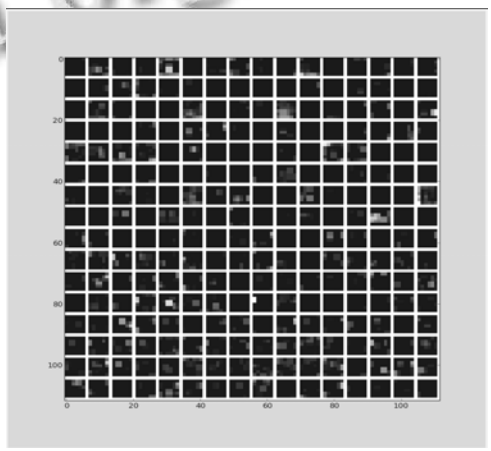

图 4 第六层全连接层网络结构

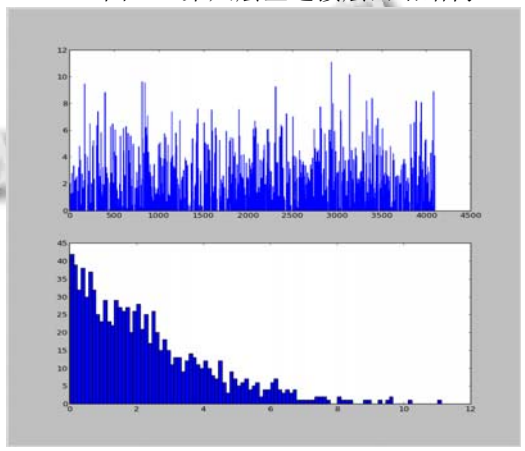

图 5 第七层全连接层网络结构

由图 6 可知，经过运算，Caffe 最后将图像变换为有语义特征的 m 维向量. 在下一章的图像检索模块正是使用的该特征向量来建立索引和做匹配计算的.

深度学习的方法很容易陷入过拟合的烦恼，为了减少过拟合，我们采用了两种数据扩展的技巧.

入训练数据, 提高模型的鲁棒性. 二是为了达到光照和颜色的不变形, 在数据集上随机增加像素的主要成分. 实验证明, 采用这些方法可以提高模型的鲁棒性, 避免过拟合.

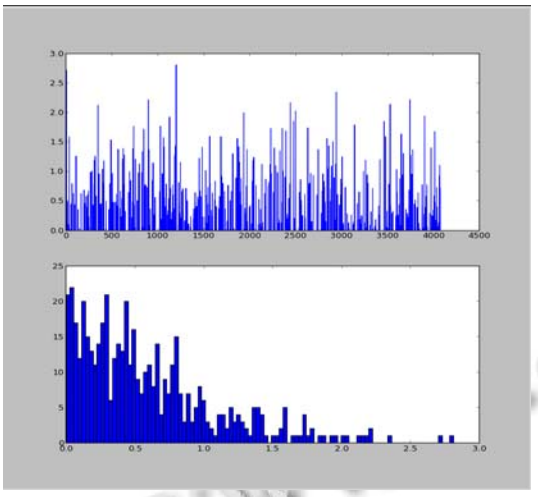

图 6 第 8 层全连接层网络结构

在 Caffe 数据层的后面是卷积层, 在第一层和第二层的后面是归一化层(normalization)和 max pooling 层, 第三层、四层的后面则没有归一化层和 max pooling 层, 第五层的后面有 max pooling 层. 在卷积层的后面是 3 个全连接层, 前面两个全连接层含有 4096 个神经元, 最后一个全连接层含有的神经元的数量由训练模型的类别数决定, 在我们实验中, 最后一层是 20 个神经元. 整个神经网络框架的参数数量大于 6000 万个. 训练出的模型就是保存的神经网络各层之间的参数, 模型占内存大小为 227M 左右. 一幅图像测试的过程就是用该图像, 通过分割, 调用 OpenCV 转化为矩阵后, 和各层网络之间的参数做矩阵相乘运算, 最后得到一个特征向量, 其中最大值的下标就是该图像对应的类别, 通过和原图像的 label 比较就可以得到分类的正确性, 进而得到测试准确率. 由于 Alexnet 网络采用了很多不寻常的技巧, 使得 Alexnet 网络比其他的深度卷积网络效果更好.

首先, 神经网络的输出函数是非线性的函数: 纠正线性单元(Relu), 而不是传统的输出函数 tanh, 在采用梯度下降法的训练方式下, 传统的输出函数的训练时间比 Relu 方式要长, 根据 Hinton 的文章[12], 我们称使用这种非线性函数的神经元为纠正线性单元(Relu), 训练模型用 Relu 作为输出单元比传统的激活函数作为输出单元快好几倍. 图 7 是在一个四层的卷积神经网络上做的测试, 网络分别使用 Relu 和 tanh 作为输出函数, 在数据集 CIFAR-10 上训练, 当达到要求的 25%的训练错误率时, Relu 比 tanh 快 6 倍, 即使用 Relu 的神经网络的学习速度更快.

第二, 采用局部响应归一化, 提高模型的泛化能力. 同时局部相应归一化可以降低模型的识别错误率, 根据 Alex 的实验, 采用这种方法可以分别降低 Top1 和 Top5 的错误率为 1.4%和 1.2%, 在 CIFAR-10 上也有 2%的提升.

第三, 采用 max pooling, max pooling 可以提高特征传输的不变性, 将不必要的特征去掉, 降低数据的维度, Caffe 中使用的是 2*2 的核, 这样可以减少 75%的数据, 同时保留最主要的特征数据, 降低了过拟合的风险.

第四, 采用 dropout, 2012 年, Hinton 在文献[9]里面提出, 训练的时候, 让一半的特征检测器停止工作可以提高模型的泛化能力, Hinton 称这种方法为 dropout.

Hinton 认为[9], 通过阻止某些特征的协同作用能够缓解模型的过拟合, 在每次迭代的时候, 每个神经元有一半的概率不发挥作用, 在下次迭代的时候又可能发挥作用, 这样可以提高模型的泛化能力, 从而降低过拟合.

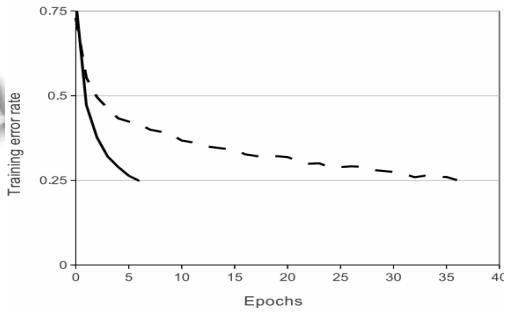

图 7 Relu 和 tanh 训练迭代次数比较曲线

1.3 基于内容的图像检索

基于内容的图像检索, 即 CBIR(Content-Based Image Retrieval)是近十年计算机视觉研究最多的领域之一, CBIR 是通过分析图像的可视化特征, 使用近似匹配算法, 从检索库中检索出一组最相似的图像, CBIR 从本质上讲是一种近似匹配技术, 它融合了计算机视觉、图像处理、图像理解和数据库等多个领域

的技术成果.

在过去, CBIR 系统主要使用的可视化特征为低级特征, 有全局的颜色特征、边缘特征、纹理特征、GIST 和 CENTRIST, 和局部的特征, 如使用局部描述子的(SIFT, SURF)的词袋模型(Bow). 传统的 CBIR 系统使用的距离匹配算法是固定的, 主要是欧几里得距离公式和 cos*ine* 相似公式.

基于深度学习的 CBIR 系统, 使用深度学习提取的特征作为索引, 实验中, 我们使用的是 Alexnet, 共有 8 层神经网络, 5 个卷积层, 3 个全连接层, 最后三层提取的是图像的高级特征, 前面 5 个卷积层提取的是图像的低级可视化特征, 实验中, 我们使用的是最后一层作为图像的特征表示, Ji Wan[1]等人的工作表明了倒数后两层作为图像的特征检索的准确率最好, 在 Alexnet 中, 最后一层为 softmax 层, softmax 是 logistic 回归模型在多分类问题上的推广, 数学表达形式如公式(1)所示, 它计算出一幅图像属于每个类别的概率, 实验中我们训练的模型有 20 个类别, 所以最后一层的维度为 20 维, 20 维的和为 1.

$$h_\theta(x^{(i)}) = \begin{bmatrix} p(y^{(i)}=1|x^{(i)};\theta) \\ p(y^{(i)}=2|x^{(i)};\theta) \\ \vdots \\ p(y^{(i)}=k|x^{(i)};\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix} \tag{1}$$

1.4 度量学习

在机器学习中, 很多算法都依赖于计算两个样本点之间的距离, 在图像检索中, 度量学习算法(Distance Metric Learning)已经被广泛的研究. 图像检索的性能不仅仅单独依赖于所提取的图像特征, 图像检索很关键的技术还在于所采用的相似度量函数. 相似度量函数直接决定图像检索的结果和检索的效率. 基于内容的图像检索与基于文本的图像检索不同, 基于图像内容的图像检索主要通过计算查询示例图像和检索库图像之间的视觉特征的相似度来决定检索的结果. 基于深度学习的图像检索, 在提取好图像的特征后, 形成特征向量, 之后基于特征向量来表征对应的图像. 在图像检索中, 判断图像之间是否相似主要是通过比较两幅图像的特征向量是否相似(距离最小)来进行的, 即把图像特征向量之间的距离比较看做图像相似度的比较, 显然, 一个好的特征向量和合适的距离度量学习算法是图像检索的关键.

2 系统概述

基于深度学习的图像检索系统, 主要使用的技术有 Python 的 Web 框架 Django, 深度学习框架 Caffe, 数据库 Mysql 等技术.

2.1 计算机系统环境

由于基于深度学习的图像检索系统使用深度学习技术, 对运算速度的要求比较高, 所以要求计算机有高性能的 GPU, 一般使用的是 Tesla K20 或者更高性能的 K40.

其他的要求如下:

系统: Ubuntu 系统 12.04 或者 14.04

CPU: Intel i3 处理器

硬盘: 200G 以上

2.2 软件开发环境

本系统使用 Eclipse 作为开发环境, 使用的 Web 服务器是 Python 版本的 Django 作为快速开发工具. Caffe 使用的编程语言是 C++(90%)和 Python(10%), 提供 Python 和 MATLAB 接口, 我们使用的是 Python 接口, 使用的编程语言主要是 Python, HTML, JavaScript.

系统需要安装的软件如下:

Cuda 驱动安装

Java 安装

Caffe 使用的相关软件的安装

Caffe 安装

Mysql 安装

Django 安装

2.3 系统的功能和性能指标

2.3.1 系统的功能要求

① 准确、快速的检索;

② 提供友好的训练模型接口(用户只要按照网页上的提示信息操作就可以训练出可靠的模型);

③ 全天 24 小时稳定工作;

④ 该平台基于开放的 B/S 架构, 具有良好的人机交互与信息展示功能;

⑤ 系统的基本信息维护功能, 主要是在系统停止工作时, 恢复系统.

2.3.2 系统的性能要求

① 系统可靠性: 达到 24 小时×7 天稳定运行;

② 检索的准确率≥80%;

③ 系统响应时间<1s;

④ 每秒检索的图像数量>10000.

2.4 系统的模块组成

本系统分为四个模块: 图像检索模块、图像检索库建立模块、模型训练模块, 系统维护模块. 系统总体结构如图 8 所示.

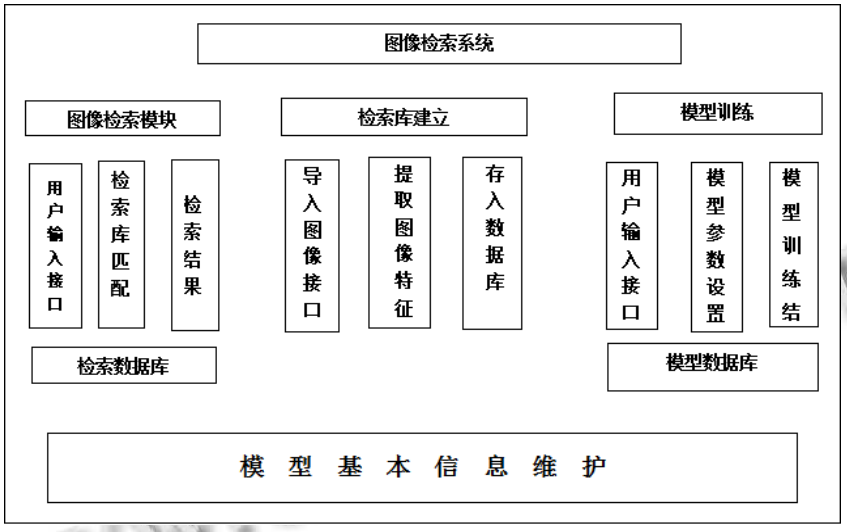

图 8 图像检索系统结构

2.4.1 图像检索模块

提取样例图像特征, 与检索库中的图像的特征向量逐一匹配, 得到检索库中每幅图像与样例图像的距离, 然后从小到大排序, 并按照用户的显示要求, 显示最靠前的结果.

图像检索的系统框图如图 9 所示.

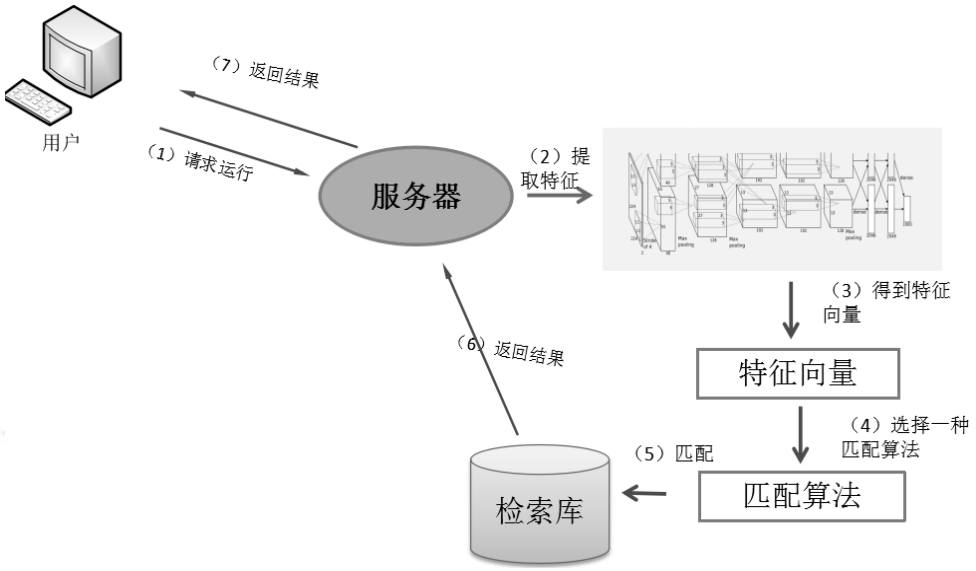

图 9 图像检索模块系统框图

步骤(2)为提取样例图像特征, 提取的方法采用深度神经网络, 经过各层网络的运算, 最后通过输出层得到特征向量, 在本文中得到的特征向量是 20 维向量. 步骤(4)为匹配算法, 本文中使用的是欧几里得距离:

$$\sqrt{(x_1-y_1)^2+(x_2-y_2)^2+\ldots+(x_m-y_m)^2}$$

步骤(5)为逐一与检索库中的图像匹配, 最后对匹配的结果(即距离)排序, 并返回一组最相似的结果.

图像检索的工作流程如图 10 所示.

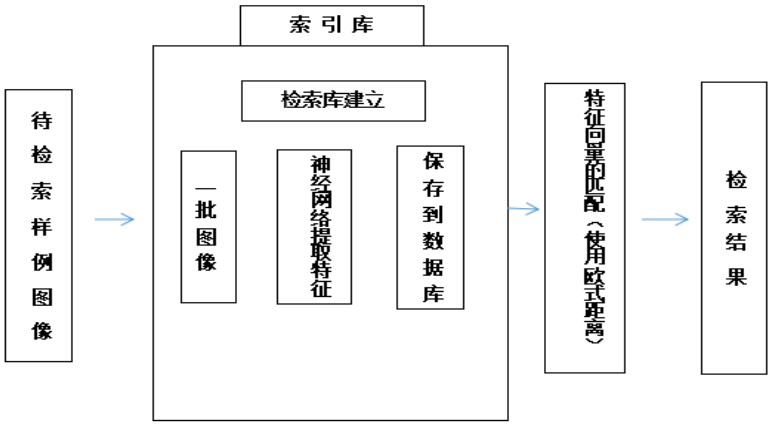

图 10　图像检索工作流程

用户在系统界面上单击“选择文件”按钮, 输入返回结果数量(例如返回最相似的100张图像), 之后点击提交(在检索操作之前, 用户要建立自己的索引库, 在图像检索预处理界面, 提交文件夹即可构建好索引库), Caffe 服务器经过提取待检测图像的特征向量, 匹配索引库, 最后返回一组最相似的结果.

2.4.2 图像检索库建立模块

检索库是图像检索系统中, 待检测图像所比较的对象, 检索库主要存储每张图像经过神经网络运算得到的特征向量.

相似度计算公式:

相似度= 1/(距离+1)

距离计算公式:

距离=$\sqrt{(x_1-y_1)^2+(x_2-y_2)^2+...+(x_m-y_m)^2}$

(m 是特征向量的维度)

这样, 当两幅图像的距离为 0 时, 相似度为 100%, 距离越大, 相似度越低.

从上面可以看出, 影响相似度计算最大的因素是图像对应的特征向量, 归根结底是训练的模型是否准确, 如果模型准确, 那么相似的两个图像, 在同一维度的值差距$(x_1-y_1)^2$ 就越小, 计算总的距离就越小, 这样得到的结果就越准确.

图像检索库建立分为两个步骤:

(1) 提取图像特征, 如图 11 所示. 神经网络框架是 Caffe, 在上一节有详细介绍.

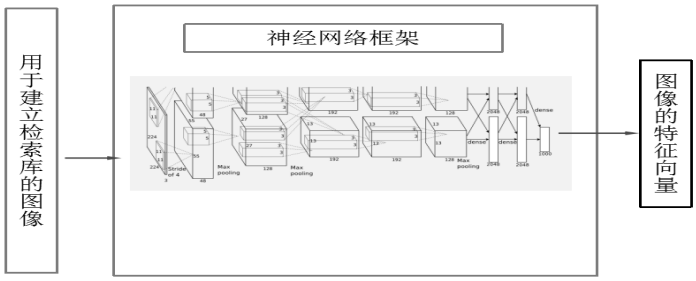

图 11　提取图像特征向量

(2) 存入数据库, 在这一步使用了一个技巧, 由于深度学习框架 Caffe 最后一层是 softmax 层, 得到的是属于每一个类别的概率, 所以, 我们根据得到的特征向量的最大一维的下标 i(index)建立数据表 tables_i, 这样数据库建立了 m 个数据表, m 对应特征向量的维数. 这样, 检索的时候, 我们根据样例图像的特征向量最大值对应的下标 i 检索对应的数据表 tables_i, 这样可以避免扫描整个检索库, 时间效率会提高约 m 倍. 存入数据库的架构如图 12 所示.

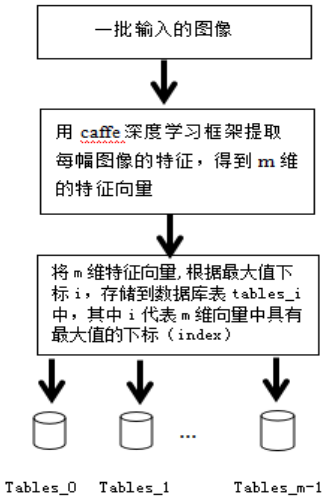

图 12 特征向量存入数据库

2.4.3 模型训练模块

用户输入一组图像(每个类别图像的数量至少大于 100 张, 类别数大于 1 类), 训练出一个对应的模型.

模型训练影响的因素: 模型训练就是训练出一个针对训练图像的模型, 最后得到的是一个二进制文件, 里面存储的是神经网络各层间的权重参数, 大小约为 227M 大小.

在模型训练中, 影响的主要因素是训练模型时间比较久, 这主要是因为神经网络的参数巨大(约有 6500 万参数), 每层之间的矩阵乘积的操作比较耗时(矩阵乘积运算大约做了上亿次的乘积运算), forward 和 backward 在每一层都做矩阵的乘积操作, 所以机器的性能对训练时间的影响比较明显, 实验中, 我们使用 GPU 并行运算来加速, 所以, GPU 的性能是关键, 我们搭建的服务器使用的是 Tesla K20c 的 GPU, Tesla K20c 速度大约是 Quadro K2100m 的 700 倍, 在性能曲线上的显示是 Tesla K20c 的机器的曲线的斜率更小. 图 13 是比较的 Quadro K2100m 和 Tesla K20c 的性能曲线.

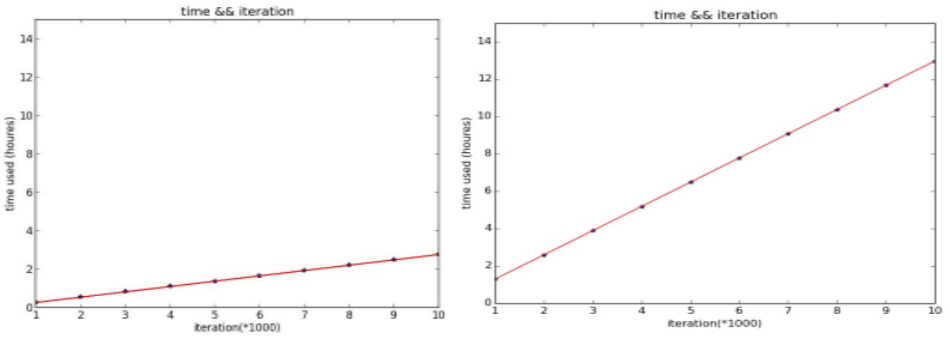

图 13 Tesla K20c 和 Quadro K2100m 训练时间对比图

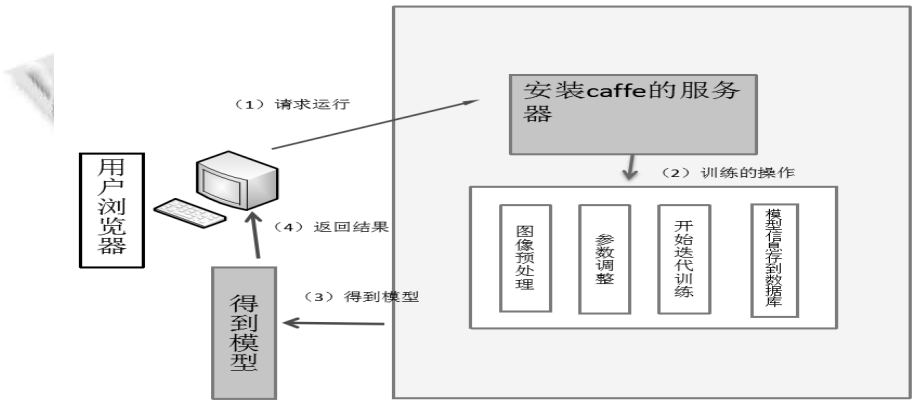

图 14 训练模型架构

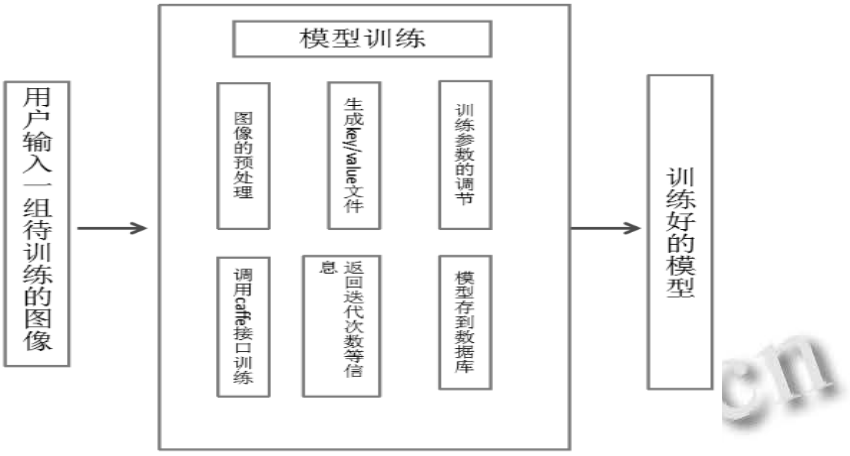

图 15　训练模型操作流程

模型训练的框架如图 14 所示, 由四个部分组成, 浏览器、Web 服务器、Caffe 服务器、数据库支持. Web 服务器将待训练的图像发送到 Caffe 服务器, Caffe 服务器经过图像预处理, 调整参数, 生成训练数据和验证数据等操作后, 调用 Caffe 训练模型接口开始迭代训练, 最后将训练的模型保存到数据库, 并反馈训练的信息到客户端, 告诉用户训练的进度.

模型训练操作步骤如图 15 所示. 用户在系统界面上提交待训练的数据, 点击提交, Web 服务器将待训练数据发送到 Caffe 服务器开始训练, Caffe 服务器实时反馈训练的进度到客户机, 告诉用户训练的进度, 最后训练结束后, Caffe 服务器返回训练好的模型的基本信息(训练人、训练时间、迭代次数、集内正确率等信息)到客户端页面.

2.4.4 系统维护模块

系统里面重要的数据库, 如模型库、检索库等, 还有服务器训练的性能(即训练的迭代次数和时间的关系)等信息, 需要提供接口供用户使用.

系统维护模块的结构如图 16 所示, 主要有三个部分, 一是索引库的清空和重建, 二是模型的删除, 三是得到系统服务器的训练性能曲线.

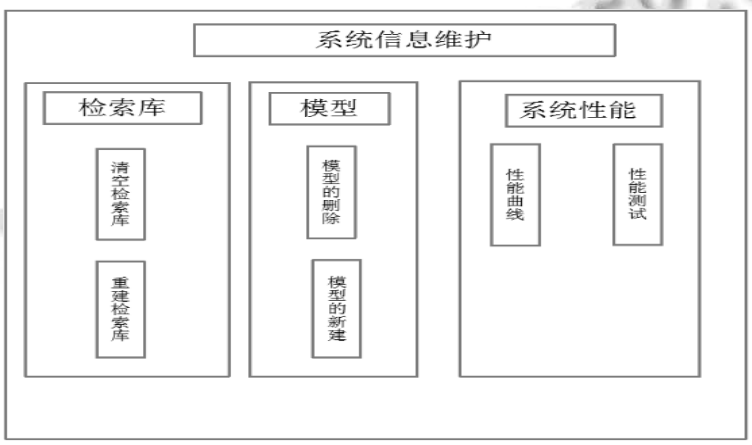

图 16　系统信息维护结构

3　实验

对于图像检索系统, 需要建立一个庞大的图像数据库, 建立索引库主要有两种方式, 在线方式和离线方式, 一般的图像检索系统在索引库的建立上使用的是离线的方式, 因为这一部分对时间的要求不高, 在特征匹配上使用的是在线计算方式, 由于这一部分和用户的交互密切相关, 所以对时间的要求和用户体验的要求都很高. 而我们整个系统使用的都是在线的方

式, 这也是我们系统的一大亮点. 我们使用 Python 的 Django Web 框架技术和 Caffe 的 Python 接口, 搭建了一个图像检索系统, 可以在线实时检索用户输入的用例图像, 同时随时可以根据用户的输入扩大索引库, 而且在用户体验和实时性上, 我们都进行了优化, 使得系统运行非常流畅.

3.1 实验平台的搭建

我们使用的服务器是有 GPU Tesla K20c 的 Dell 工作站, 服务器训练一个有 20000 张训练素材的模型(fine-tune 的方式)大概需要 3.5 小时的时间. 系统搭建的步骤如下:

① 安装部署 Caffe, 并简单测试;

② 安装部署 Djando 和数据库;

③ 基于前台和后台开发程序.

3.2 结果

在模型训练上, 我们使用的训练集大小为 20000 张图像, 验证集大小为 6000 张图像, 得到的模型信息如表 1.

为了评估训练模型是否可以用于图像检索, 我们使用在图像检索中广泛使用的特定范围的准确率(P@K)来测试, 实验中, 我们的检索库大小为 20 万张图像. 测试结果如表 2.

表 1 模型信息

epoch 数	30
验证准确率	88.44%
类别数	20

表 2 测试结果

类别数量	P@K	准确率
20 个类别	P@K=1	0.86
	P@K=5	0.85
	P@K=10	0.80

在时间性能上, 由于我们将检索库按照每幅图像的特征向量最大值的下标 *i* 映射到了 K 个表中, 所以, 与其他实验相比, 我们的检索效率提高了接近 K 倍的速度.

下面是图像检索排在最靠前的结果: 检索前十张的平均相似度为 80%以上.

由表 2 可以看出来, 检索一张的准确率和模型分类的准确率数值相近, 因为, 检索依据的距离(欧几里得距离)主要由样例图像特征向量的最大值决定, 该最大值即为该图像分类到该维代表类别的概率, 所以, 检索一张的准确率和模型的分类准确率相近.

图 17　实验结果对比图

4　总结

在本文中, 我们只使用了 8 层神经网络, 训练的样本数是每个类别 1000 张, 从测试的结果可以看出: (1)深度学习可以从原始的图像中学习到高层的语义特征; (2)训练的模型具有很好的鲁棒性, 对于网上下载的图像, 检索的结果准确率都很高(大于 80%); (3)深度学习是唯一的端到端的系统, 中间不需要人为的参与, 不需要先验知识, 特别适合处理海量数据. 从现阶段看, 深度学习技术是处理语义鸿沟最有前途的技术, 同时, 我们也发现, 深度学习在图像检索中的发展方兴未艾, 未来有着巨大的空间, 在图像检索中正趋向使用更大更深的网络结构, Alexnet 只包含了 5 个卷积层和 3 个全连接层, 而 GoogleNet[10]的网络结构超过了 20 层, 更深的网络结构使得反向传播更加困难. 与此同时训练数据的规模也在迅速增加. 这些都迫切需要研究新的算法和开发新的并行计算系统以更加有效的利用大数据训练更深的模型.

参考文献

1 Wan J, Wang DY, Hoi SCH, Wu PC, Zhu JK, Zhang YD, Li JT. Deep learning for content-based image retrieval: A comprehensive study. Proc. of the 22nd ACM International Conference on Multimedia. ACM. 2014. 157–166.

2 Hinton GE, Osindero S, Teh YW. A fast learning algorithm for deep belief nets. Neural Computation, 2006, 18(7): 1527–1554.

3 Rumelhart DE, Hinton GE, Williams RJ. Learning internal representations by error propagation. Nature, 1986.

4 Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems. 2012. 1097–1105.

5 Deng J, Dong W, Socher R, Li LJ, Li K, Li FF. ImageNet: A large-scale hierarchical image database. IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009). IEEE. 2009. 248–255.

6 Nair V, Hinton GE. Rectified linear units improve restricted biltzmann machines. Proc. 27th International Conference on Machine Learning (ICML-10). 2010. 807–814.

7 Donahue J, Jia YQ, Vinyals O, Hoffman J, Zhang N, Darrell ET. DeCAF: A deep convolutional activation feature for generic visual recognition. ICML. 2014. 647–655.

8 Breiman L. Random forests. Machine Learning, 2001, 45(1): 5–32.

9 Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv: 1207.0580. 2012.

10 Szegedy C, Liu W, Jia YQ, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. Proc. of the IEEE Conference on Computer Vision and Pattern Recognition. 2015. 1–9.

11 Donahue J, Jia Y, Vinyals O, et al. DeCAF: A deep convolutional activation feature for generic visual recognition. Computer Science, 2013, 50(1): 815–830.

12 Nair V, Hinton GE. Rectified linear units improve restricted boltzmann machines. Proc. 27th International Conference on Machine Learning (ICML-10). 2010. 807–814.