南京理工大学

NANJING UNIVERSITY OF SCIENCE & TECHNOLOGY

《媒体计算基础》
课程论文

课程名称：《媒 体 计 算 基 础》

指导老师：李 泽 超

学生姓名：肖 林 航

学 号：919106840638

目录

使用 VGGNET16 实现以图搜图

--基于内容的图像检索(CBIR)

姓名：肖林航　　学号：919106840638

摘要：经过了数周的《媒体计算基础》课程学习，我对图像检索的相关理论知识有了更为深入的理解，对一些常用的技术，例如颜色特征提取、SIFT 特征提取及相关检索方式，有了初步认识，也进行了一些实践验证。通过比较多种检索方式，最终我选择了 VGGNET16 模型实现基于图像的图像检索功能。

关键词：VGGNET16；CBIR；图像检索；深度学习

一、前言

基于内容的图像检索可大致分为两类：一类是传统的提取图像的颜色、纹理、形状轮廓等底层特征进行检索；另一类是近些年兴起的基于深度学习框架的图像检索方法。无论是哪种方法，基于内容的图像检索的基本流程都是相似的：首先对数据集进行预处理和图像特征的提取；接着对提取到的特征进行归一化等处理后存储到特定文件中；最后选取图片，提取特征后使用检索引擎进行相似度匹配，检索出最为匹配的若干张图像。

在完成课程作业的过程中，我也尝试了使用基于颜色特征和 SIFT 特征的图像检索。对于颜色特征检索，我使用了颜色直方图和卡方距离等方法，在检索过程中，需要大量的计算和比较，当数据集很大时，其检索性能会大大下降。对于 SIFT 特征检索，在特征提取时会占用较大的内存，且在特征列表堆叠和聚类过程中会消耗大量的计算机资源，耗时很长。（**性能较低的旧版代码尝试：** *https://github.com/Nuyoah-xlh/CBIRBySiftAndColor*）总而言之，基于颜色特征和 SIFT 特征的图像检索在特征提取和检索中会有不同的问题，当数据集规模达到 50000 张以上时，它们的检索性能和检索效果都不太令人满意。

经过一番学习和实践，最终我选择了使用 VGG-16 模型进行图像检索。VGG 模型是牛津大学的 Visual Geometry Group 提出的。其本质是增加网络的深度来提高网络的性能。VGG-16 模型的使用也非常简单，他是一个预训练过的模型，不需要从头开始训练，这使得基于 VGG 的迁移学习变得相对容易。下面我将给出我对 VGG-16 的理解和使用心得：

二、VGG-16 模型理解

2.1 深度学习的理解

在我看来，深度学习就是通过大量数据的数据训练模型，构建多层的人工神经网络，将海量的数据通过非线性方法提取权重，最终构造出一个具有一定“智慧”的算法，它能够对训练集以外的数据做出合适的处理。

CNN(卷积神经网络)是一种深度学习模式，因其由卷积层等结构组成而得名，而 VGG-16 又是 CNN 的一种经典模型，其具体结构及理解将在下文展示。

2.2 VGG-16 模型结构

VGG-16 主要包含 3 种层次，具体为：13 个卷积层、5 个最大池化层和 3 个全连接层。其中只有卷积层和全连接层具有权重系数，两者一共有 16 层，这也是 VGG-16 中 16 的来源。

卷积层用来提取输入图像的特征，可多次进行卷积，迭代提取更为高级的特征。池化层用来缩小模型的大小，提高特征提取效率。全连接层可看作使用矩阵向量乘积将一个特征空间线性变换到另一个特征空间，通常在最后进行特征加权，方便进行特征分类，优化计算性能。

VGG-16 的基本流程可概括为：预处理图像，使得输入一张尺寸 224*224*3 的彩色图像；接着使用 64 个卷积核进行两次卷积，一次最大池化层处理，图像大小变为 112*112*64，即通道数变为 64；接着使用 128 个卷积核进行两次卷积和一次最大池化层处理，图像大小变为 56*56*128，即通道数变为 128；同理使用 256 个卷积核进行三次卷积和一次最大池化层处理变为 28*28*256，即通道数变为 256；再使用 512 个卷积核进行三次卷积和一次最大池化层处理变为 14*14*512；再重复上一步骤，得到 7*7*512；最后经过三次全连接层处理得到 1000 维的结果向量。其结构可用以下的经典图形表示：

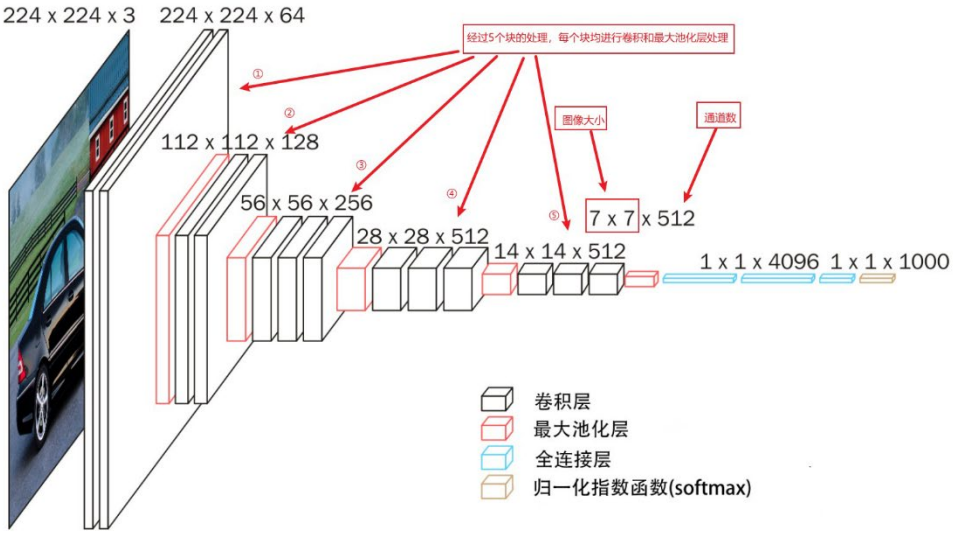

表 2.1-VGG16 结构图

可以看出，VGG-16 模型可分为 5 个块的处理，卷积的深度逐渐变深，提取到的信息越来越细化，可归纳其特征提取流程如下：

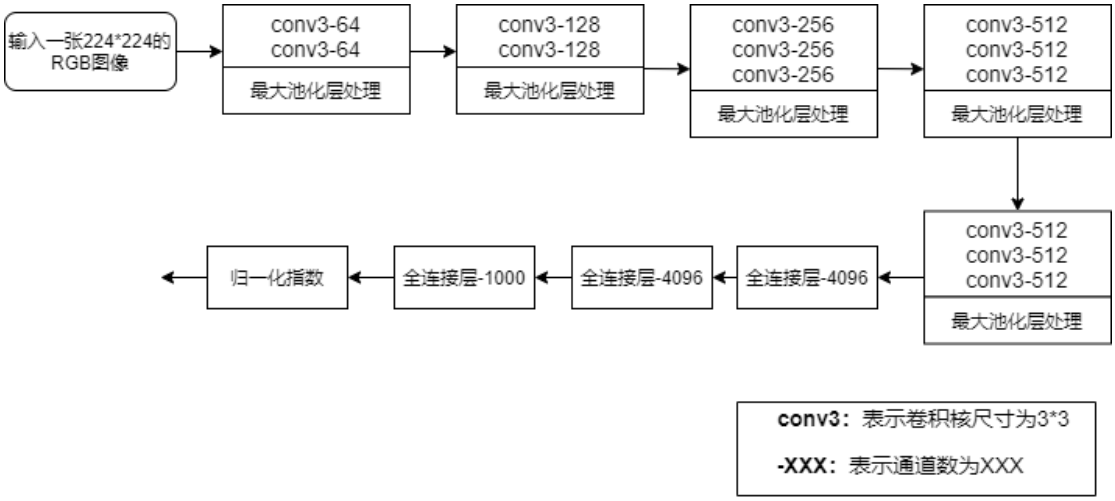

表 2.2-VGG16 特征提取流程

2.3 卷积核及相关层

VGG-16 的小卷积核是其一个重要特点，VGG-16 模型采用了多个 3*3 的卷积核而不是传统的 5*5 或 7*7 卷积层，采用小卷积核能够获得更深的网络层次，从而拥有更有效的图像信息。VGG-16 一般需要输入 224*224*3 方阵大小的图像，令输入矩阵大小为 w，卷积核大小为 k，补零层数（填充值）为 p，s 为步长，w’为输出矩阵的大小，则计算卷积后的特征图像大小的公式为：

$$w' = \frac{w+2*p-k}{s} + 1。$$

经过计算很容易发现，两个 3*3 的卷积核堆叠和一个 5*5 卷积核产生的感受

野（输出图像上的像素点在输入图片上映射的区域大小）相同，且相比后者，两个 3*3 卷积核能够减少一定的参数量，更节省资源。

池化层主要用来缩小模型的尺寸，令 w’为输出图像的尺寸，w 为输入图像的尺寸，p 为池化层尺寸，s 为步长，则其处理图像矩阵后的尺寸计算公式为：

$$w' = \frac{w - p}{s} + 1$$

2.4 softmax 层

一般情况下，模型最后输出层的节点数与分类目标数相等，例如节点数为 n，则神经网络会产生一个 n 维的数组，在理想情况下，他可能会出现形如[0,0,0,0,1,0,0…0]的数组，1 为类别对应节点，但实际上我们常常希望能够同时兼顾到较大和较小的值，这时我们就可以使用 softmax 函数对其进行复杂的加权和与非线性处理，最终得到形如一组概率值，其和为 1，形如[0.05,0.08,0.5,…0.1]。令有一个数组 X，Xi 表示 X 中的第 i 个元素，那么求 Xi 的 softmax 值为：

$$Si = \frac{e^{Xi}}{\sum_j^n e^{Xj}}$$

损失函数是用来衡量模型预测好坏的函数，它能够表现预测数据与实际数据的差距程度。那么，损失函数值越小，则代表模型的预测效果越好。在 VGG-16 中，采用交叉熵损失作为损失函数：

$$L_i = -\log\left(\frac{e^{f_{y_i}}}{\sum_j^n e^j}\right)$$

模型预测、获得损失和学习的流程大致为：VGG 模型神经网络最后一层得到一组得分；将该组得分使用 softmax 函数处理获得一组概率值；接着将概率输出与相关 one-hot 编码进行交叉熵损失函数进行计算。

2.5 相似性度量

图像检索的重要一环就是图像的相似性度量，常用的方法就是计算样本之间的“距离”，而这种“距离”的计算也有很多种，例如欧氏距离、马氏距离、汉明距离、夹角余弦距离。在本项目中，我采用了夹角余弦距离作为相似性度量函数。夹角余弦可以来衡量向量方向上的差异，设两个 n 维样本点 a(x1,x2,...,xn)和 b(y1,y2,...,yn)，可以使用类似于夹角余弦的概念来衡量它们间的相似程度：

$$cos\theta = \frac{a \cdot b}{|a| \times |b|}$$

该公式展开可得：

$$cos\theta = \frac{\sum_{k=1}^{n} x_k y_k}{\sqrt{\sum_{i=1}^{n} x_i{}^2} \sqrt{\sum_{j=1}^{n} y_j{}^2}}$$

夹角余弦的绝对值范围为[0,1]。夹角余弦越接近 1，表示这两个向量方向越接近，当两个向量的方向重合时夹角余弦取最大值 1。

三、代码实现

VGG-16 模型的预训练需要大量的计算机资源和时间耗费，而 Keras 的应用模块 Application 为我们提供了带有预训练权重的 Keras 模型，VGG16 就是其中之一，且能够兼容 Tensorflow。因此使用 Keras 提供的 VGG16 框架可以省略前期繁杂的训练，只需要关注后期的数据图像处理即可。

3.1 创建 VGGNet 类

首先创建一个类，用于初始化 VGG16 模型，其中主要有两部分组成：重写 __init__()函数和提取特征向量。重写__init__()函数时，规定其输入形状为 224*224*3，使用在 ImageNet 上预训练过的权重和最大池化层，接着调用 keras.applications 的 VGG16()函数返回 VGG16 模型，并设置概率预测。

get_feat ()函数用于提取特征向量，函数有一个 img_path 参数作为待提取特征的图像路径。首先调用 keras.preprocessing.image 的 load_img()函数，按照 224*224 的目标尺寸加载图像；接着将其转为 Numpy 数组并展开便于分析计算；然后使用 keras.applications.vgg16 的 preprocess_input()函数进行预处理，包括像素缩放、RGB/BGR 转换等处理；接着进行预测，返回图像属于每一个类别的概率；最后将 features 经过 L2 范数归一化处理后返回。

3.2 创建 getFeatures 类

创建一个 getFeatures 类用于提取图像数据集所有图像的特征向量并存储到.h5 文件中。首先获取所有图片路径，接着使用上一步创建的 VGGNet 类中的 get_feat ()函数获取它们的归一化特征向量，最后将所有的图片相对路径及其特征向量存储到 index.h5 文件中。

3.3 创建 search 类

创建一个 search 类用于提取被检索图像的特征向量，并检索出数据集中与其最匹配的若干张图像。首先使用 VGGNet 类中的 **get_feat ()**函数得到归一化特征向量，接着使用 **numpy.dot()**函数计算余弦相似度，然后使用 **numpy.argsort()**实现了线性搜索。最后可返回相似度最高的若干张图像的存储路径和相似度得分。

3.4 其他

此外，还建立了 app.py 及 SelectAndSearch.py 文件，主要用来创建可视化界面程序，接收图像数据并显示。

四、问题及解决方案

4.1 传统方法的尝试

在设计初期，我尝试了使用传统的特征提取方式，包括颜色特征提取和 SIFT 特征提取。但都遇到了问题，对于颜色特征，能基本实现功能，但匹配的效果较差，更重要的是，当数据集规模达到数万时，检索性能会大大下降；对于 SIFT 特征提取，检索性能和检索效果有所提高，但提取数据集特征的过程，数组堆叠和聚类会将占用大量资源，当提取五万张图像时，会导致电脑内存不足使程序崩溃。

为了能够顺利提取更多的图象特征，我尝试使用多个.csv 文件分别提取数据集特征，最后再分别从多个文件中对比检索，但最终的检索效果会有所下降。

最终经过学习研究，我尝试了使用基于深度学习的 VGG-16 模型进行特征提取，经过实践发现，该模型在图像检索方面有诸多优点，无论是检索性能还是检索效果，都远远超过了传统方式。

4.2 检索性能

首先我尝试了使用冒泡排序的思想进行检索，但其检索耗时还是不太令人满意，5000 张图像时耗时近 10s，当数据集规模超过五万张时，甚至会无响应。

经过学习和测试，我利用 numpy 的 argsort()函数，通过矩阵的点积等方法实现了更快速的线性检索。经测试，同样的机器环境下，在规模为五万张图像的数据集下检索图像，耗时稳定在 1s 左右，相比之前，有了很大的性能提升，性能问题也得以解决。

五、运行情况

5.1 选择被检索图像

运行 app.py 后，选择需要检索的图片，一次仅可选择一张，如图所示：

表 5.1-主界面

5.2 检索

选择图像后，点击“搜索”即可显示搜索结果，几张检索结果图如下：

表 5.2.1-检索结果 1

表 5.2.2-检索结果 2

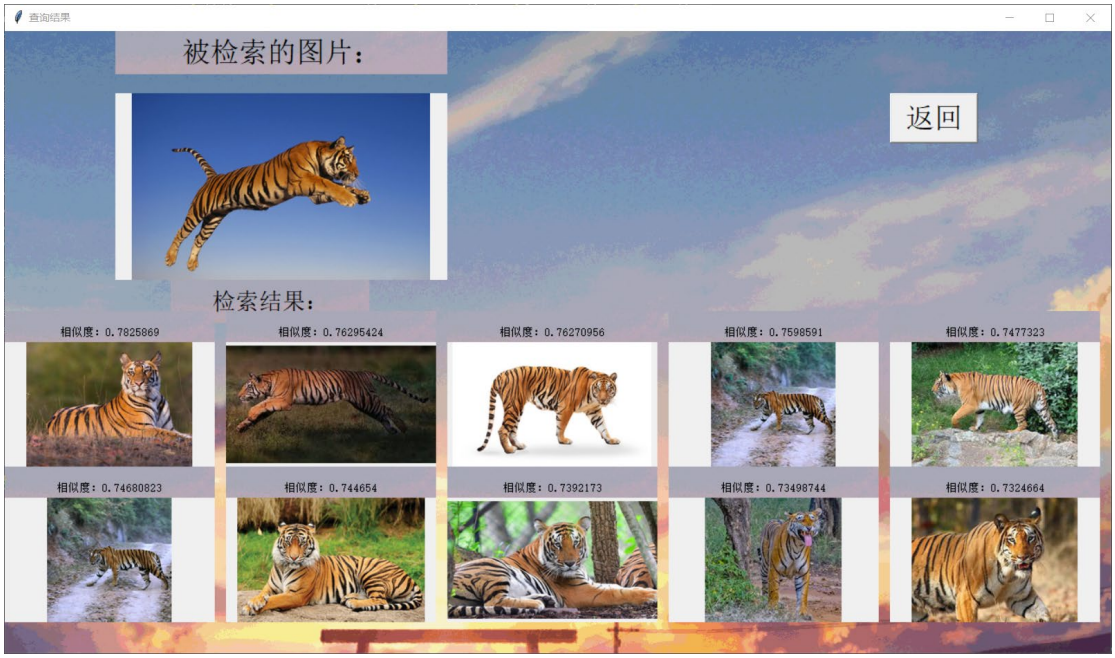

表 5.2.3-**检索结果** 3

六、结语

经过不断的尝试，发现并解决问题，最终项目的实现效果总体令人满意。从最初的提取颜色特征到提取 SIFT 特征，再到最后的使用 VGG-16 卷积神经网络提取特征，图像检索的性能和效果不断提升。

VGG-16 能够取得很好的检索效果，与其自身结构的设计有关。它是一个典型的串型神经网络，使用 3*3 的小卷积核多次卷积，从而获得层次更深的神经网络。且每次卷积或池化后得到的图像的各区域与原图是对应的，即方向位置一致，这使得提取的特征也包含了被检索图像的位置信息，检索出的图像也会更加精准。

尽管 VGG-16 模型已经能够取得很不错的检索效果，但依然存在着很多不足。VGG-16 模型的参数多达 1 亿多，十分臃肿，随着深度学习的持续发展，更为精准、高效的模型层出不穷，例如 GoogleNet、ResNet 等。神经网络的发展日新月异，基于深度学习的图像处理拥有广阔的前途，我们仍需不断学习，不断探索。

《媒体计算基础》这门课程理论与实践相结合，在课上我学到了很多关于各种媒体形式的底层理论知识，通过课下自学一些 VGG-16 模型的知识，最终实现了基于内容的图像检索，在这过程中我学到了很多，也对图像处理萌生了兴趣。总而言之，这门课令我受益匪浅！