Image Retrieval

图像检索实验报告

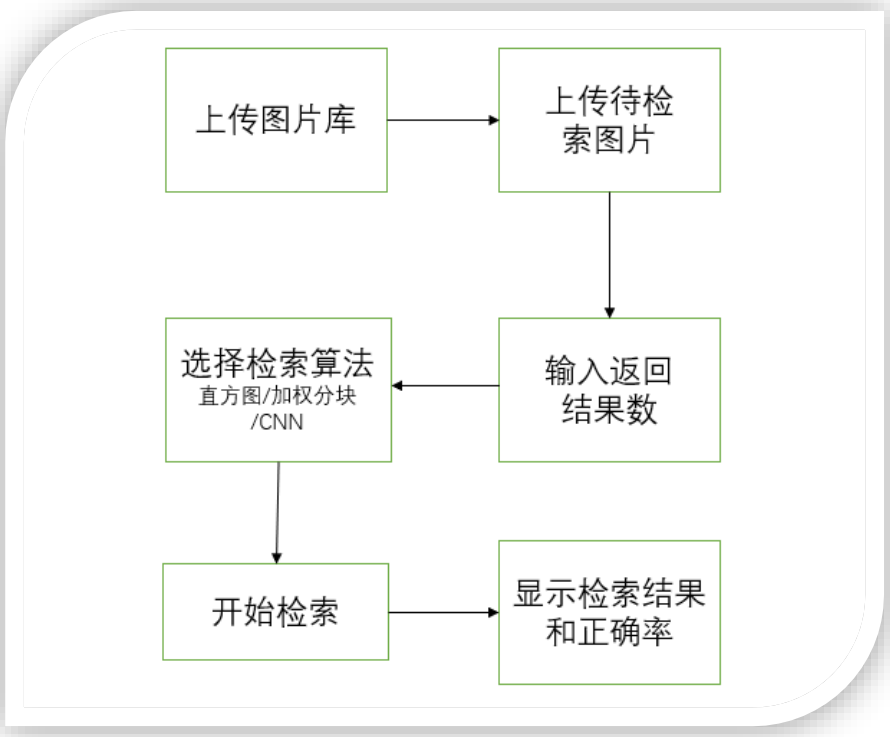
初阳 10165102158
李智慧 10165102117

目录

需求分析

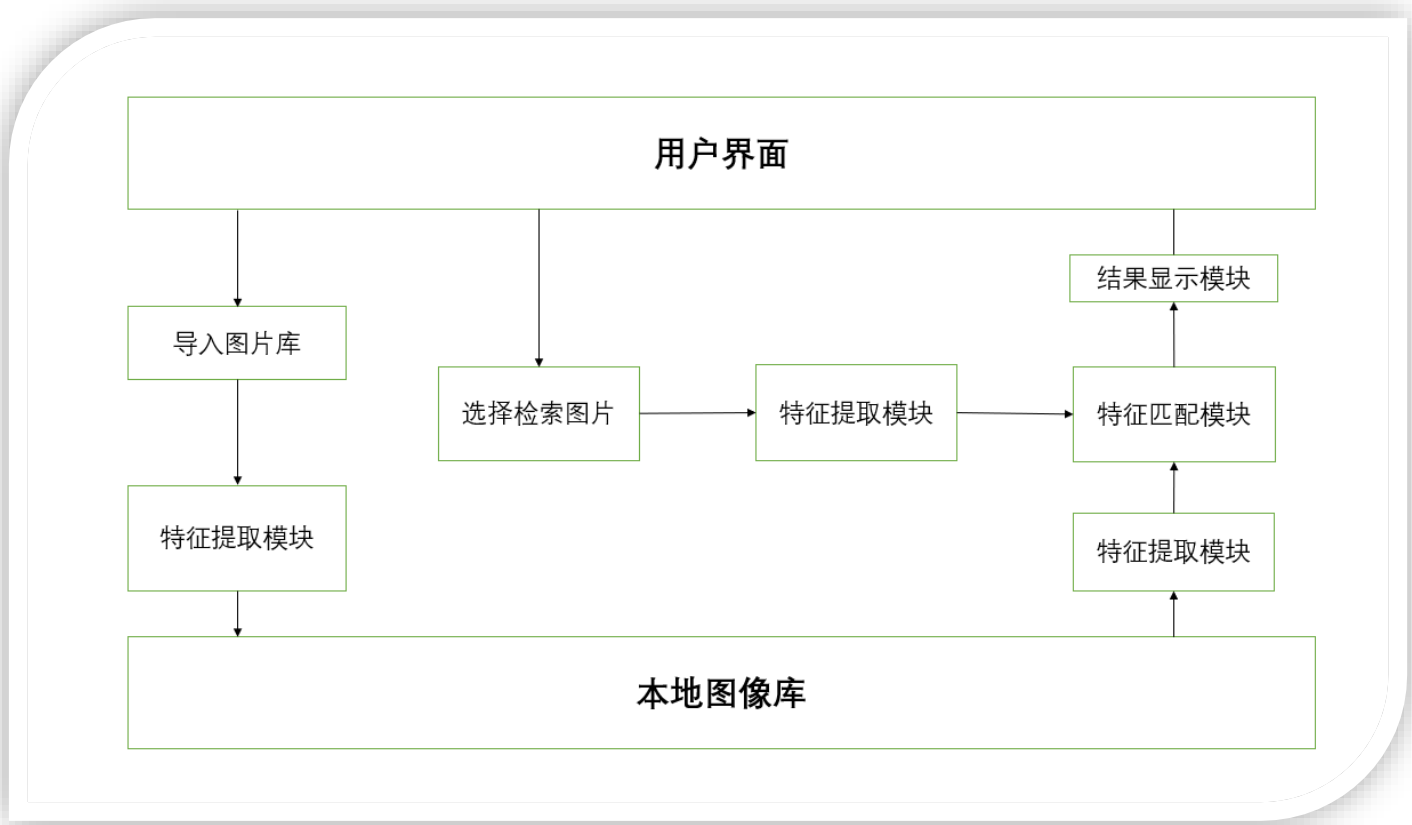
1. 基于颜色直方图的检索
2. 基于图像分块加权检索
3. 基于 CNN 的检索
4. 正确率计算
5. 图片标签识别和用户修正（未实现）
6. 检索结果图片的查看与保存
7. 良好的用户交互界面

我们设计出如下过程:



功能简介

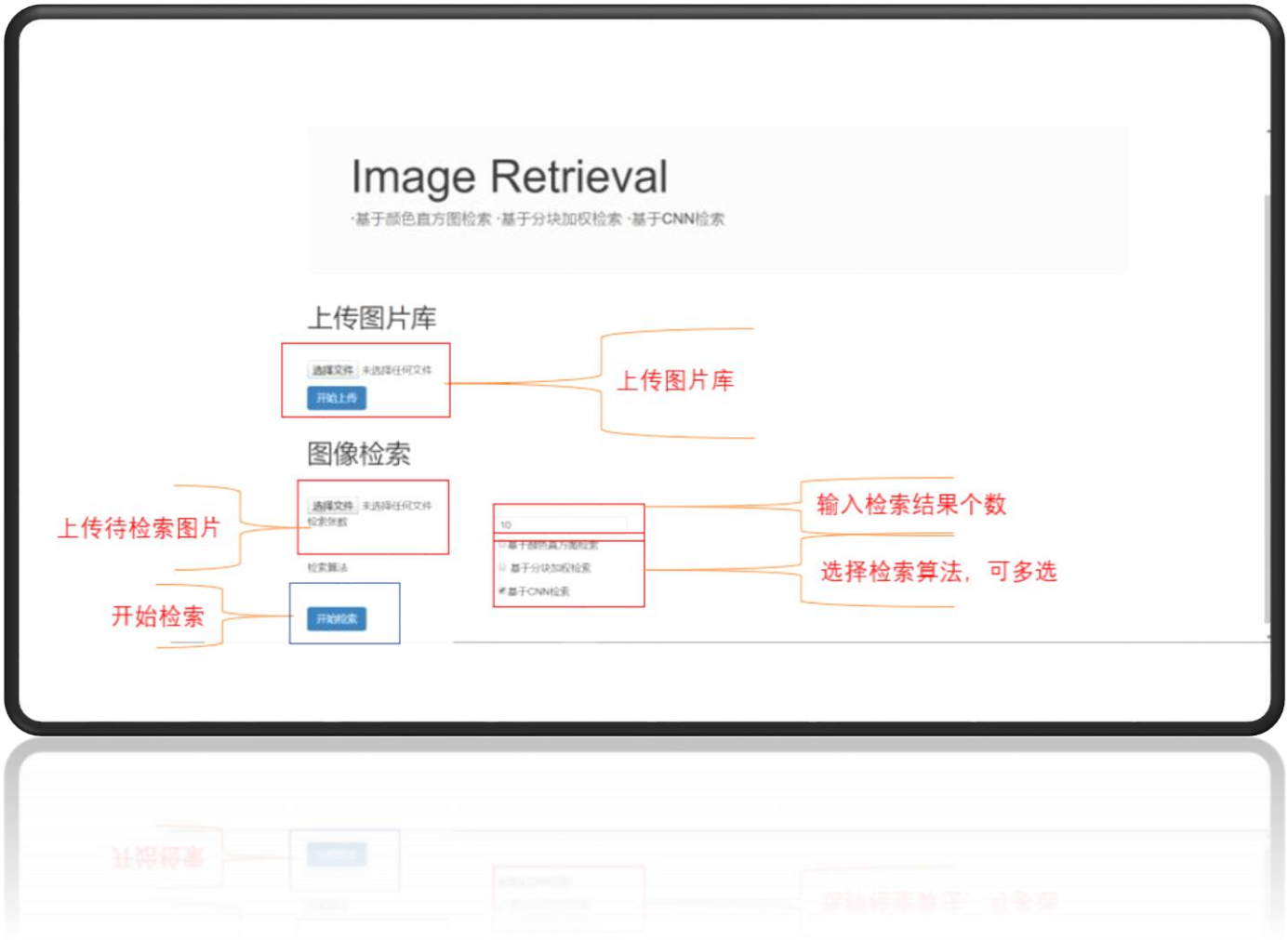
系统框架图



界面展示

1. 初始界面展示

上传图片库，上传待检索图片，输入检索结果个数，可同时选择多个检索算法，开始检索。



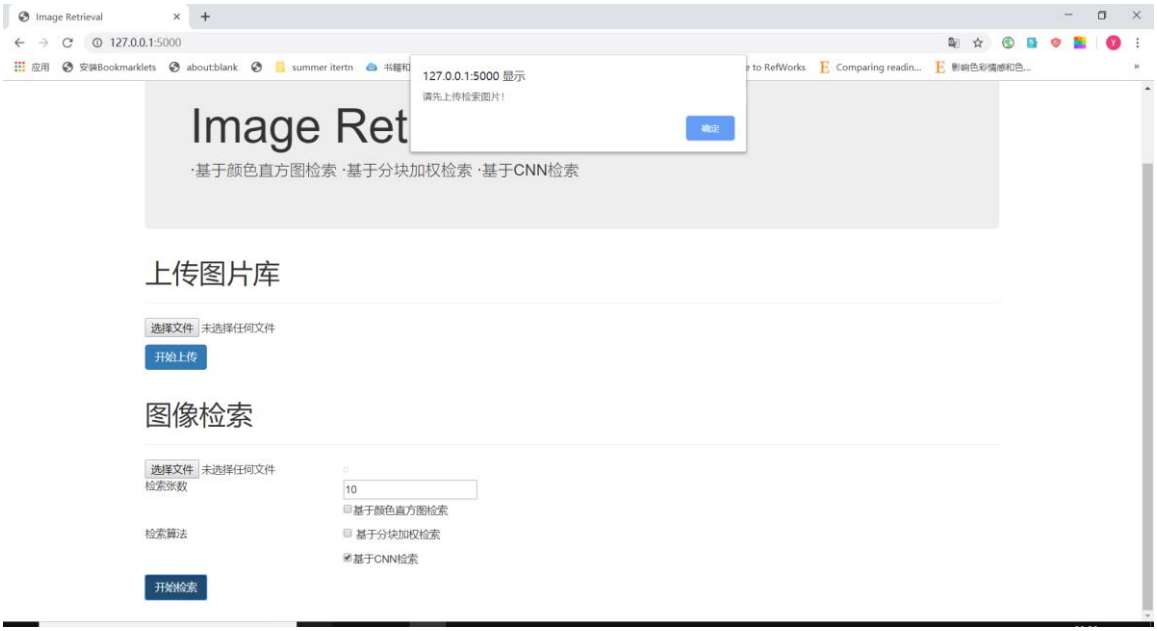
2. 搜素结果界面展示

- 显示检索结果和命中率
- 可同时显示选择的多个检索算法的结果
- 可通过左右滑动来查看检索结果

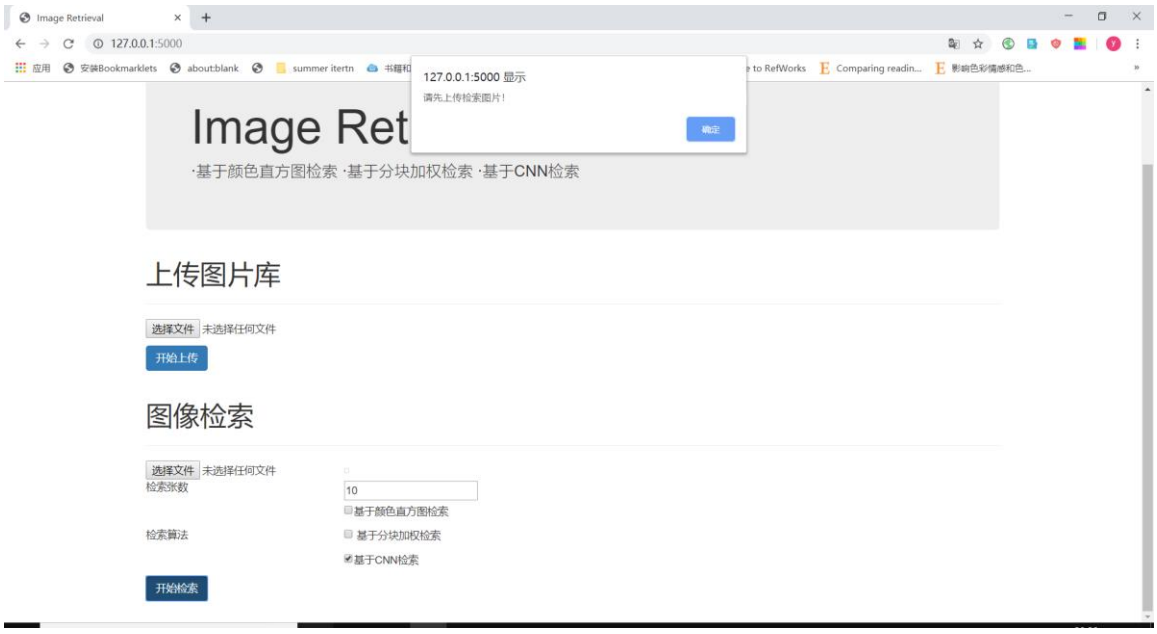
- 检索结果大图查看

3. 操作错误提示界面

- 未上传图库提示界面

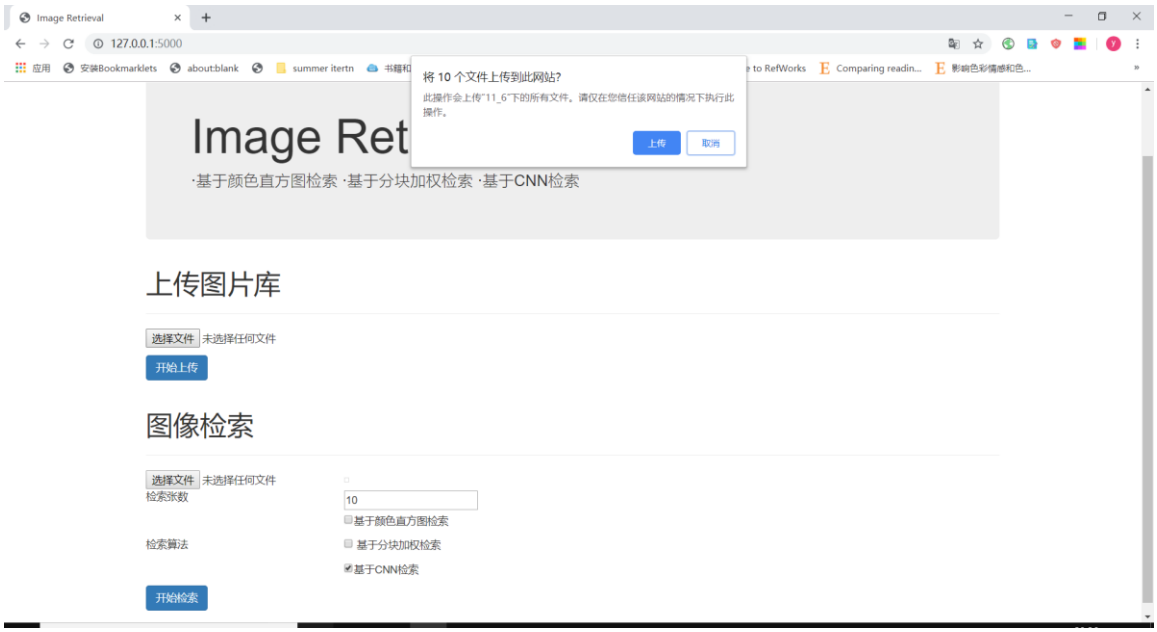


- 未上传待检索图片提示界面

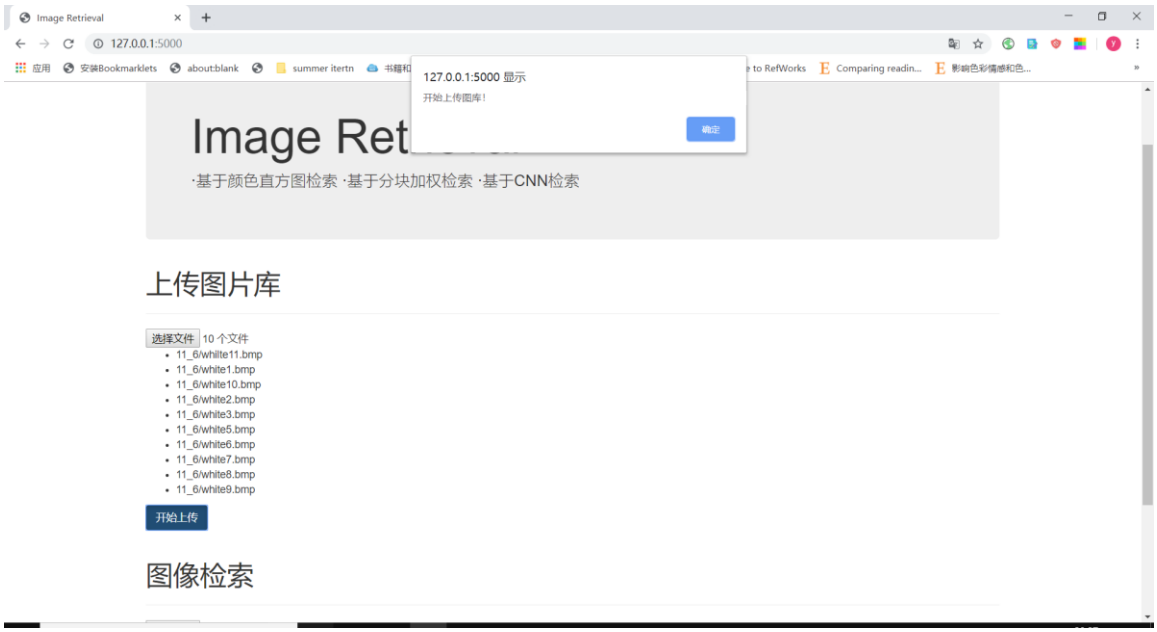


4. 上传和检索提示

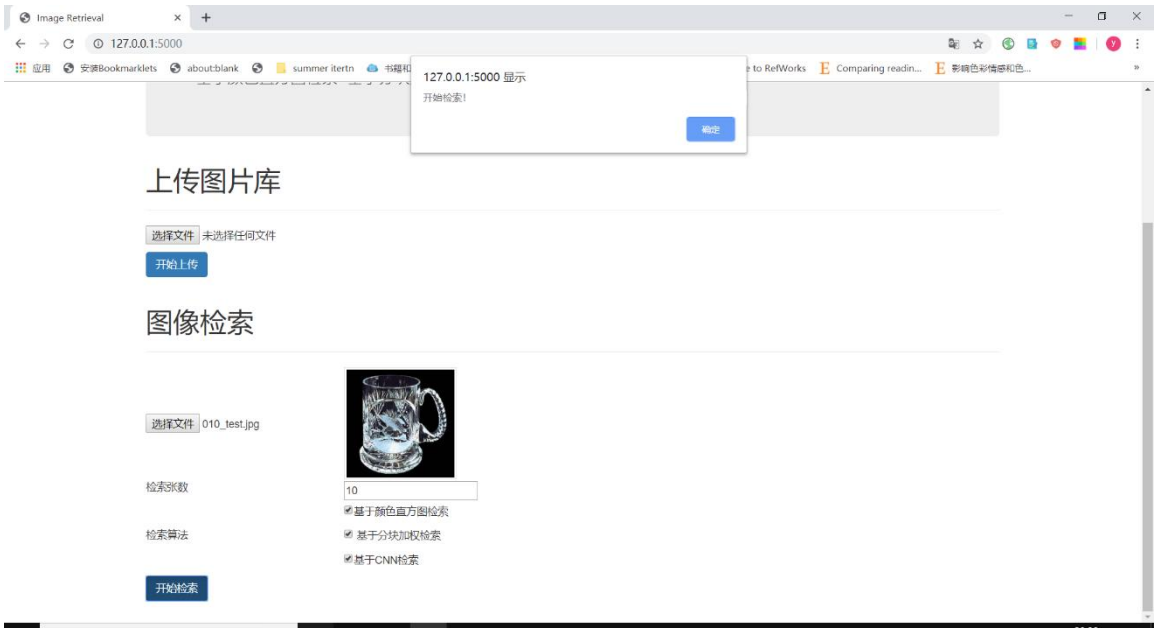
- 是否上传到网站提示



- 正在上传提示



- 正在检索提示



算法流程图

基于颜色直方图的检索算法

概述

基于直方图的颜色检索过程主要包括颜色模型选择、直方图提取及量化和直方图的相似性度量，以下介绍具体算法的实现过程。

[1] 颜色特征提取

本次实验采用 HSV 颜色空间，图像处理时，一般采用 HSV 模型。HSV 颜色空间与 RGB 颜色空间不同的是它的 H,S,V 是没有关联的，这个特点更加符合人眼的视觉特性。H 为色调（Hue），S 为饱和度（Saturation），V 为亮度（Value）。将 RGB 模型转化为 HSV 模型的公式如下：

$$V = \frac{(R+G+B)}{\sqrt{3}}$$

$$S = 1 - \frac{\sqrt{3}min(R,G,B)}{V}$$

$$H = \begin{cases} \theta & G > B \\ 2\pi - \theta & G \leq B \end{cases}$$

$$\theta = \cos^{-1}\frac{1/2[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}}$$

[2] 直方图特征提取和量化

a) 得到图像后，先将其由 RGB 颜色空间转化为 HSV 颜色空间。

b) 根据人的眼睛的辨别颜色的能力，对 H,S,V 分量作非等间隔量化。
本实验将 H,S,V 分量分别做 16 级，4 级和 4 级的非均匀量化。

c) 构造特征矢量。将 HSV 颜色空间的 3 个颜色特征分量合成一维特征分量。具体方法如下：

$$G = HQ_SQ_V + SQ_V + V$$

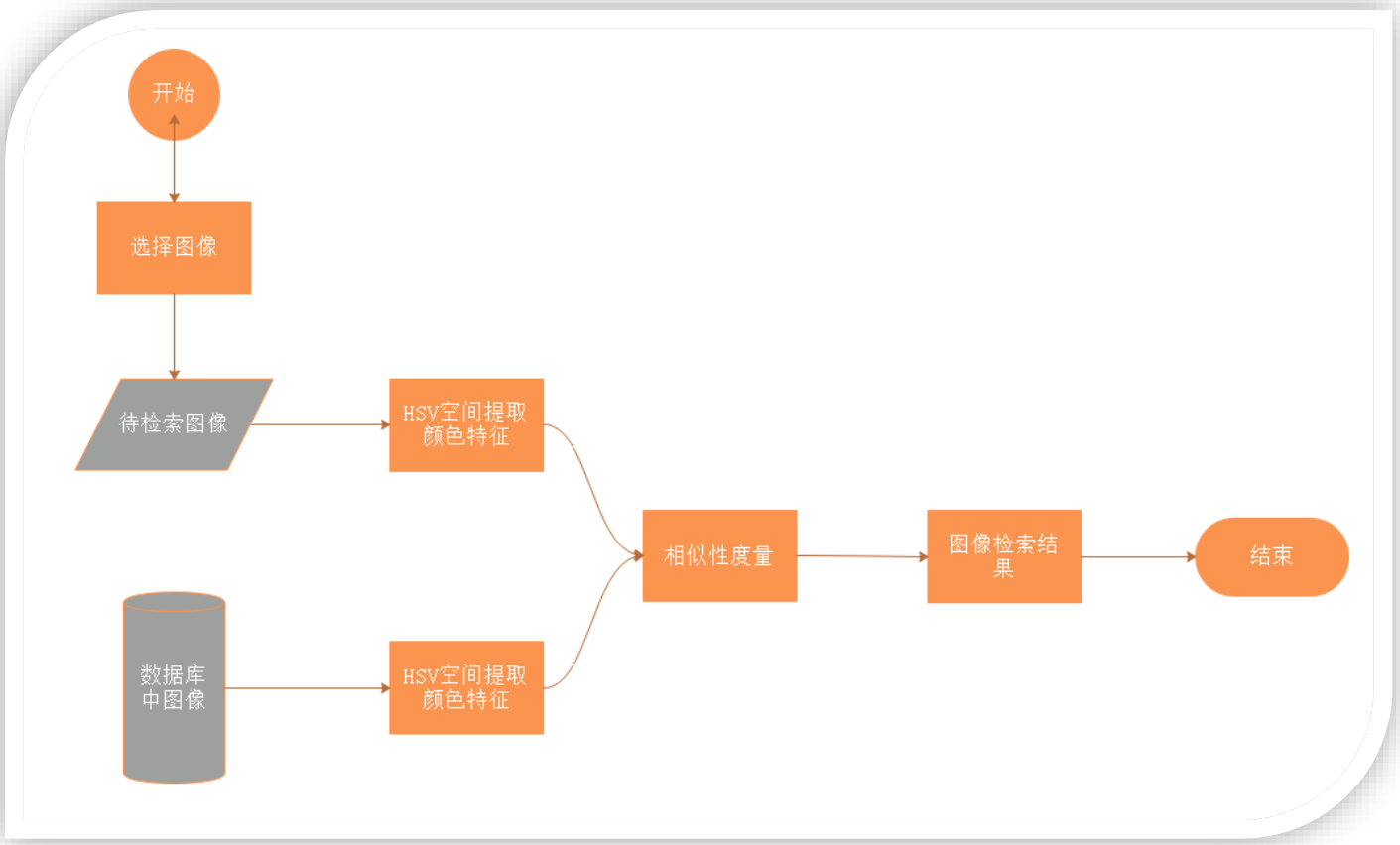
Q_S 和 Q_V——分别表示 S 和 V 的量化级数，本实验取 4

G——表示 HSV 颜色空间的 3 个颜色特征分量，取值为【0,255】

[3] 相似性度量

欧式距离和皮尔森相关系数为常用的相似性度量方法。
先进行 100 张图片的相似度比较，两种算法的查准率分别为 0.35 和 0.38。所以，采用皮尔森相关系数作为之后的两幅图像之间的相似性度量。

算法实现流程图



基于分块加权的检索算法

概述

对图像进行均匀分块处理，对各分块分别提取主颜色特征，通过对图像的兴趣点进行计算，统计各分块区域内兴趣点数目的占比，以此作为该分块的权值并得到加权后的颜色特征。兴趣点的分布是基于图像内容决定的，通过不同分块中兴趣点数占比赋予权值更能体现出不同图像之间主体内容的分布差异。此种方法不仅能够增加图像的空间信息，同时将图像各区域敏感程度不一的影响因素也考虑在其中。

以下介绍具体算法的实现过程。

[1] 图像分块

将图像均匀划分为 3*3 个子块。

[2] 分块图像的颜色特征提取和量化

对分块后的单个图像，采取上一方法中的算法，进行颜色特征提取和量化。在颜色直方图的基础上，提取主颜色（出现频率大于 5%的颜色）构成颜色自己，这个子集所包含的向量就表示为图像的主颜色特征 f。

主颜色指的是图像中某些出现频率交大，对图像的内容表达起主要作用的颜色。

[3] 兴趣点提取

对分块后的单个图像，提取 Harris 兴趣点，得到图像兴趣点总个数 n，然后分别统计 9 个子块中各自所包含的兴趣点个数 d_i(i=1,2,3…9)，从而求出每个子块在图像特征提取时所占据的权值 $W_i=d_i/n$

[4] 图像颜色特征提取

对图像划分的每一个子块分别提取特征 f_i(i=1,2,3…9)，然后根据上一步求得的各分块所占权重 W_i，即可得到图像分块加权后的特征向量

$$F = \sum_{i=1}^{9} (W_i \times f_i)$$

[5] 图像纹理特征提取

纹理特征能够反映出物体表面和周围环境两者之间存在的相关性，体现图像自身灰度变化的专属特性。

a) 采用灰度共生矩阵来对图像的纹理特征进行描述。
灰度共生矩阵的生成是对图像上间隔特定距离的两个像素点之间所具有的某种灰度状态统计生成的，充分体现了图像的空间特性

b) 利用图像的灰度共生矩阵求出图像的能量，对比度，熵值和自相关四个分量，将这四个分量进行综合即可得到图像的纹理特征向量

$$S = [ASM, CON, ENT, COR]$$

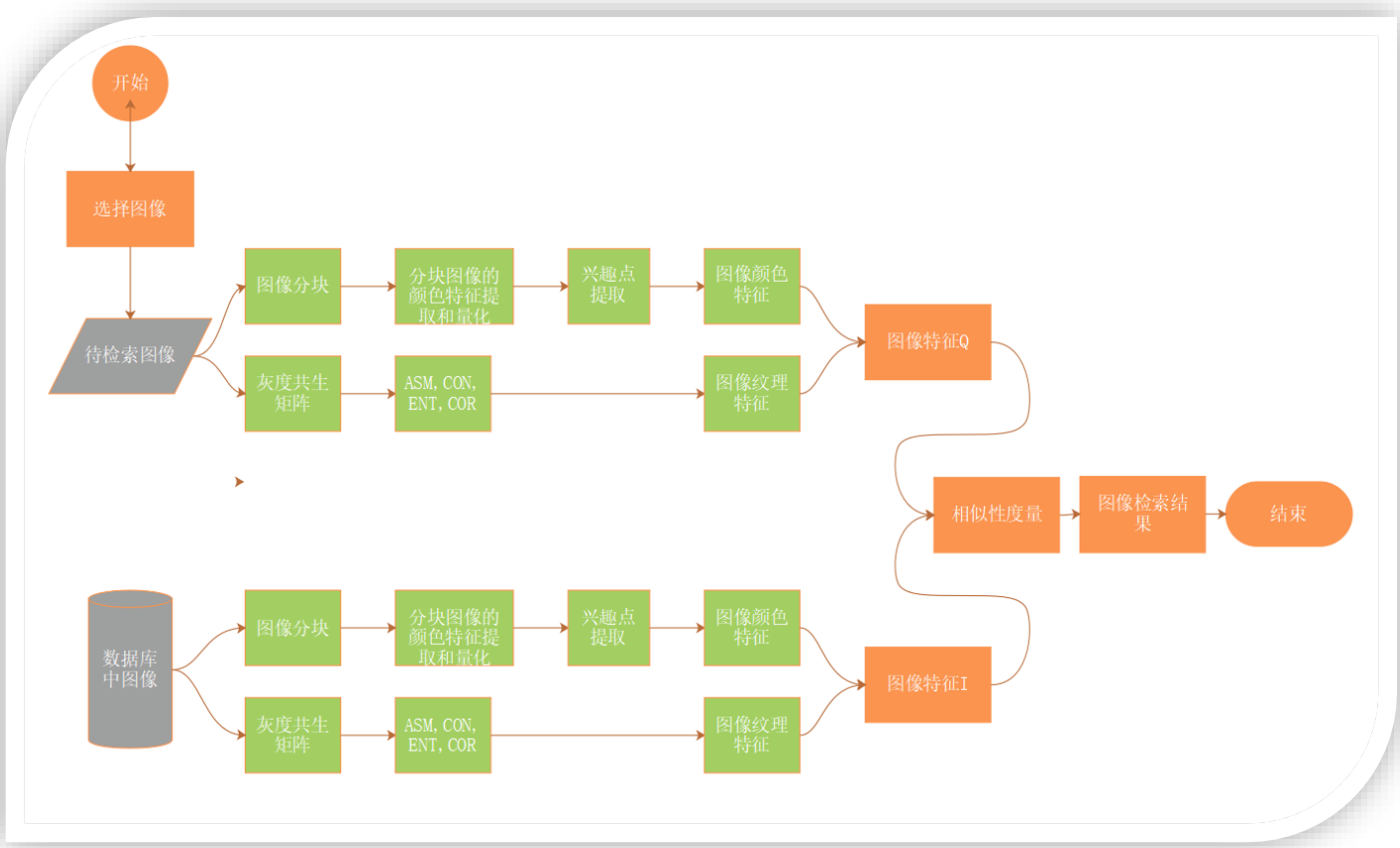
[6] **图像相似度匹配**

Q 表示待检索图像，I 表示图像库中的图像

$$S(Q, I) = W_C S_{COLOR}(H_Q, H_I) + W_T S_{TEXTURE}(G_Q, G_I)$$

W_C和W_t分别表示颜色特征和纹理特征两者在进行综合处理时各自的权值大小，本实验在特征相似性融合时将颜色和纹理赋予了同等的权值。

算法实现流程图

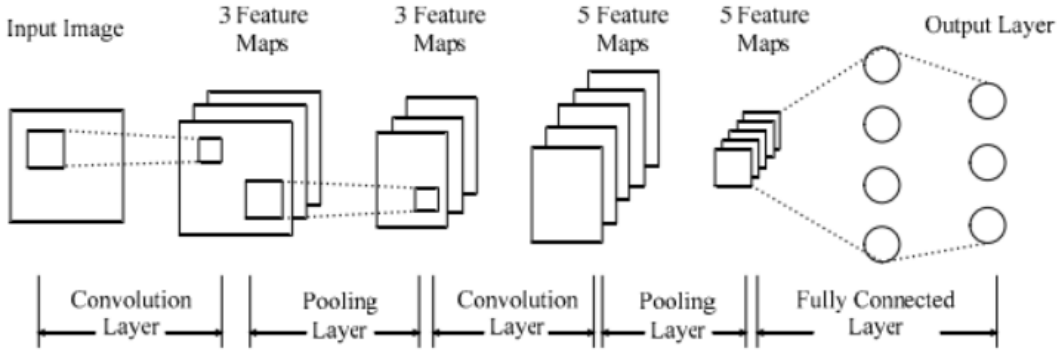


基于卷积神经网络的检索算法

概述

卷积神经网络（Convolutional Neural Network，CNN）是深度学深在计算机视觉

领域的一个应用，是人工神经网络的一种。因为神经网络的通用性，一个网络不仅可以用来实现图像分类，也可以实现图像检索。卷积神经网络的基本原理是：给定一个图像库，给定一个卷积网络模型，通过优化一个代价函数来训练最优权值，并通过最优权值来提取图像特征的过程。CNN 的基本结构包括：输入层，卷积层，下采样层，全连接层和输出层。下图是一个卷积神经网络模型结构图：



本实验使用 keras 库中已封装好的图像处理模型 keras.applications.vgg16 。

以下介绍具体算法的实现过程。

[1] 提取图像库的特征

用 A={a1,a2,a3…an}表示图像库，该库有 n 张图像组成。

图像库每一张图像的特征表示为 AH={H1,H2,H3…Hn}。

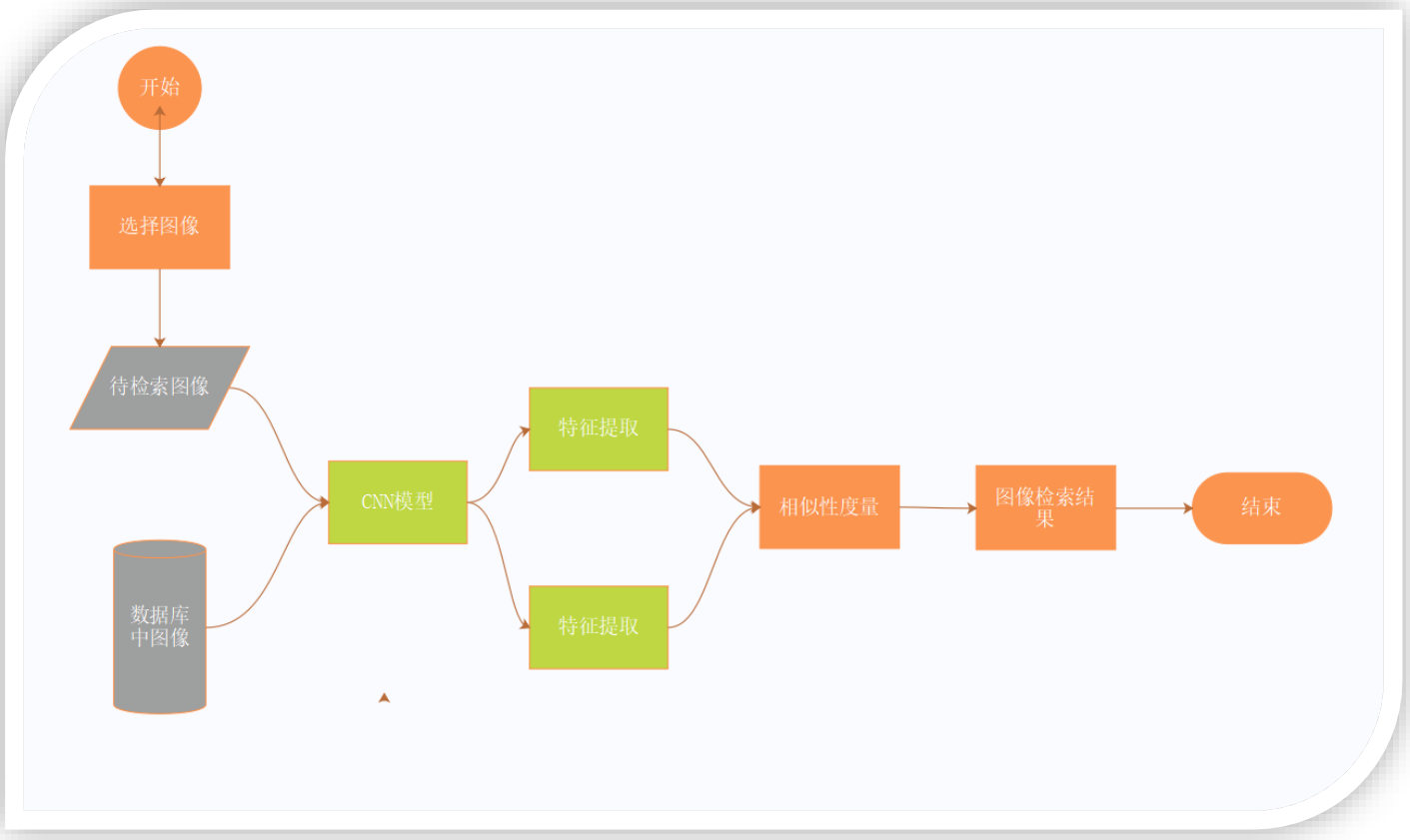
[2] 提取待查询图像 *Q* 的特征

对给定一张查询图像 Q，输入卷积神经网络模型，提取与图像库对应的层次输出 Hi 作为该图像的特征

[3] 图像相似度匹配

分别依次计算待检测图像 Q 的 CNN 特征向量和图像库中各个图像的 CNN 特征向量之间的距离，当计算结果越小时那么该图像与待检测图像就越相似

算法实现流程图



实验结果和分析

实验环境

flask 0.12.2+Python 3.6.8

数据量

数据库图片数量：3494

数据库图片类别：10

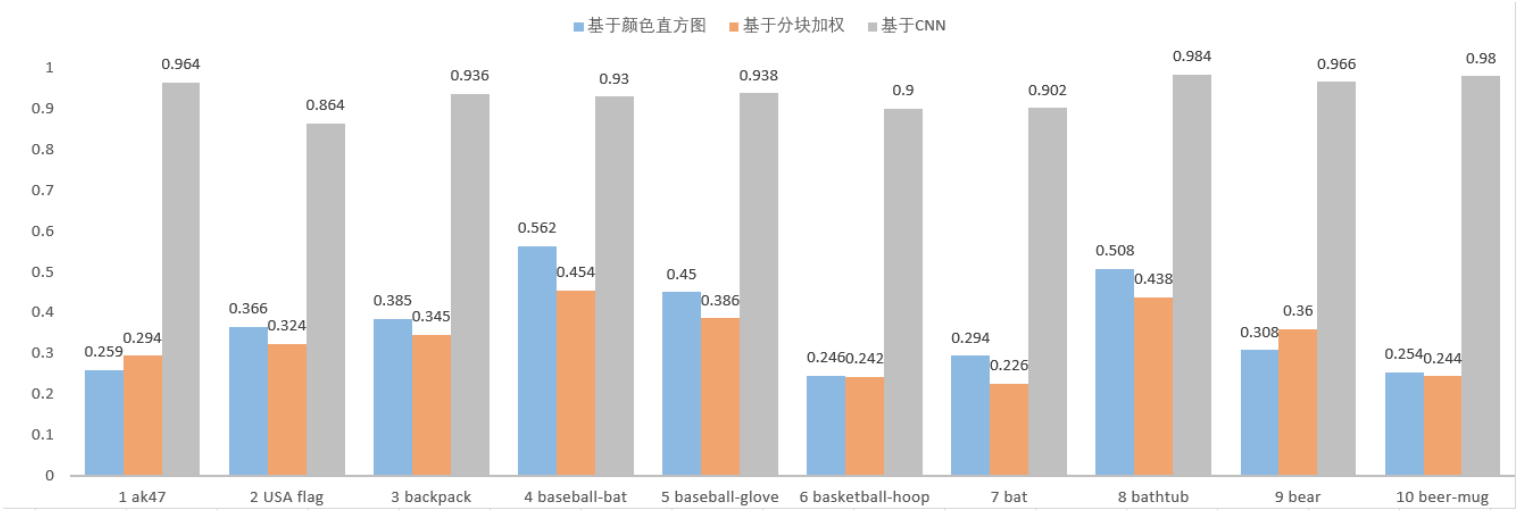
算法准确率

在实验算法完成后，对三种算法，十类图片分别进行准确率实验。结果如下：

	基于颜色直方图	*基于分块加权*	*基于 CNN*
1 ak47	0.259	0.294	0.964
2 USA flag	0.366	0.324	0.864
3 backpack	0.385	0.345	0.936
4 baseball-bat	0.562	0.454	0.93
5 baseball-glove	0.45	0.386	0.938
6 basketball-hoop	0.246	0.242	0.9
7 bat	0.294	0.226	0.902
8 bathtub	0.508	0.438	0.984
9 bear	0.308	0.36	0.966
10 beer-mug	0.254	0.244	0.98
	0.3632	***0.3313***	***0.9364***

对三类检索算法，检索算法准确率最高的类使用荧光黄标出，检索算法准确率最低的类使用蓝色字体标出。

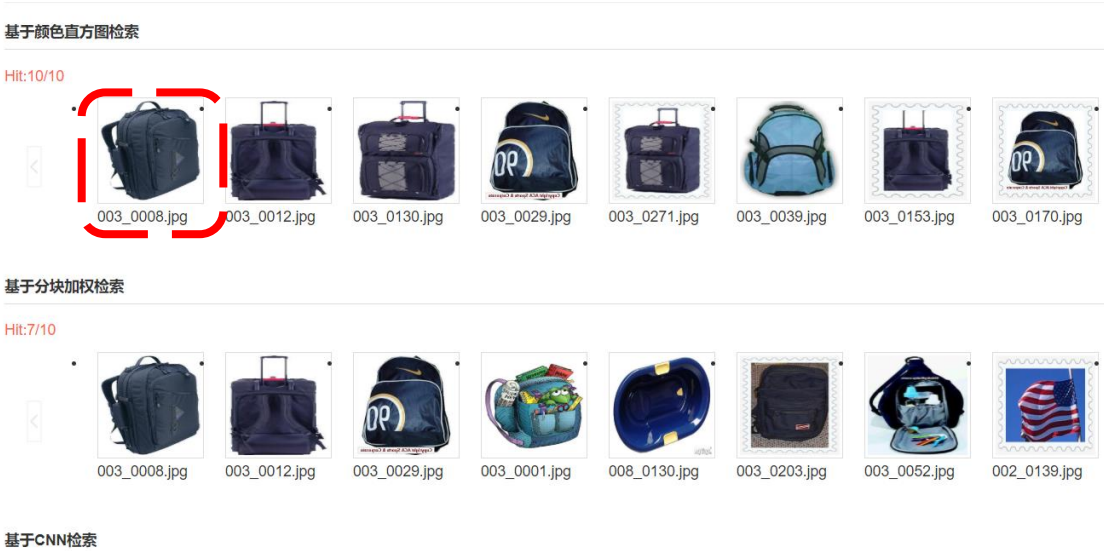
由表格可知，基于平均正确率，基于 CNN 的图像检索算法效果最好，其次是基于颜色直方图的检索算法，最后是基于图像加权的检索算法。将结果可视化下效果如下：



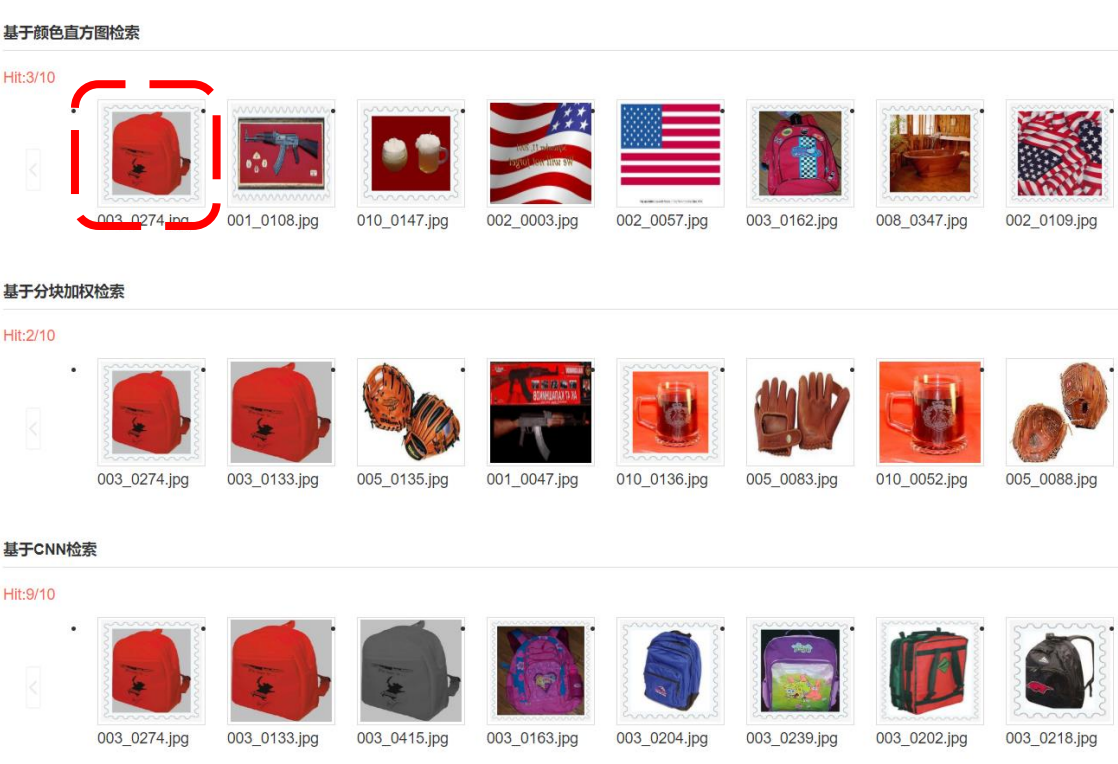
测试图片结果

将测试图片进行测试，得到结果如下，被检索图片使用虚线圈出：

①



②



③

基于颜色直方图检索

Hit:4/10

基于分块加权检索

Hit:3/10

基于CNN检索

Hit:10/10

④

基于颜色直方图检索

Hit:1/10

基于分块加权检索

Hit:1/10

基于CNN检索

Hit:8/10

⑤

基于颜色直方图检索

基于分块加权检索

基于CNN检索

⑥

基于颜色直方图检索

Hit:2/10

基于分块加权检索

Hit:3/10

基于CNN检索

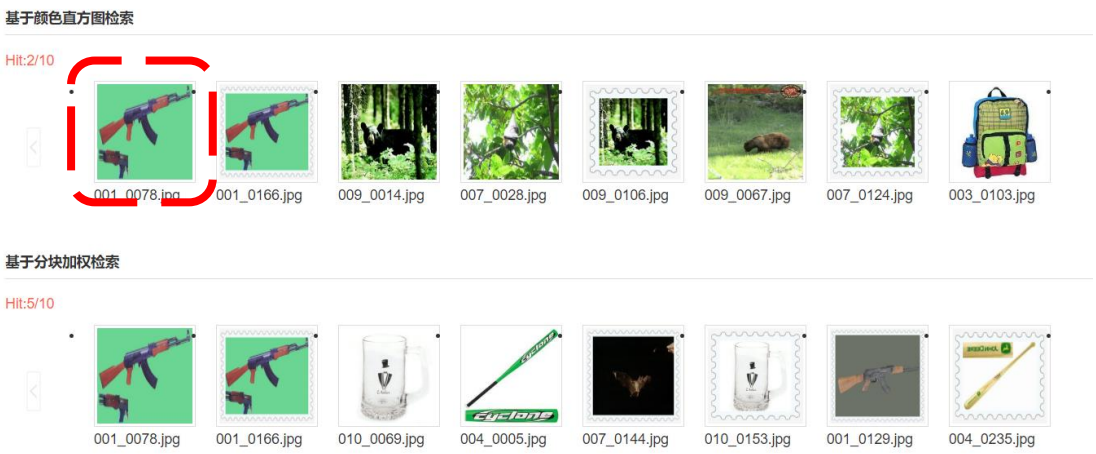
Hit:10/10

结果分析

[1] 基于 CNN 的图像检索算法，各方面表现都最优

[2] 由结果可得出，三种算法对只存在单一物体，纯色背景的检索图像表现较好。对于检索背景较为杂乱，表现则较差。以测试用例④为例，待检索目标为棒球棒，但是待检索图片上还存在人脸，蓝天等。分析基于 CNN 检索算法的结果，这种算法还将人脸、红帽子等信息作为检索目标。

[3] 关于基于颜色直方图检索和基于图像分块加权检索，虽然后者没有前者检索正确结果正确率高，但不难看出后者在轮廓处理上的优越性。以测试用例⑥为例，后者的第四个用例，虽然颜色和待检索图片不同，但轮廓形状十分相似，而前者却没有检索出这个图片。为了更好地说明二者之间的差别，我使用了下面这个用例：可以看到，基于颜色直方图检索，只能检索出绿色和黑色的颜色信息，而基于图像分块加权检索，还能检索出“条状”这个特征。



遇到的问题和改进

1. 关于用户界面，有一个输入框，来输入检索张数，发现当输入值为负数，小数，和超过图片库大小的值，都会报错。所以，进行输入检验，输入值只能为正整数。
2. 关于用户界面，当我未上传图片时，误点击开始检索，出现程序报错，报错为 Internal Server Error。但普通用户是不知道这句话的含义的。所以，为了防止误点击行为，我设置了图片检索前的检验操作，判断图片是否为空，图片格式是否有错等，以达到比较好的用户交互体验。弹窗示例如下：

3. 关于运行速度，我们知道程序的运行速度非常影响用户体验。虽然我的实验程序在检索时运行速度不算慢，但我想了两个方案来提升用户体验。一是，优化程序，提高程序的运行速率，二是，在前端设置进度条，给用户一个实时的反馈。但是二者的实现难度均较高。
4. 关于功能，实验开始之前，我们对需求进行了分析，并预想增添除基本功能之外的其他功能。直到实施时，才发现实现的困难，特别是前端功能。这也确实是我们知识的欠缺，还需要继续学习。
5. 关于导入图片库，因为我们的系统使用网页来实现，所以服务器无法获得导入的图片库在本机的路径，所以只能实现图片都单张上传到服务器，再统一导入。

总结

优点：界面简洁大方，用户交互好，检索结果准确

缺点：程序的鲁棒性较弱，可能还有未测试到的错误情况；很多想实现的功能未能完成；程序的运行速度还需要提高

分工：

初阳：UI 和检索算法代码实现，报告撰写

李智慧：原型设计，报告撰写

收获：

本次大作业让我们对多媒体的理解更近了一步。在实验的过程中，我们逐渐了解到了用户交互的重要性，以及从最初的原型设计到系统的实现，都需要我们投入精力，用心设计，一点点打磨，精益求精。

本次实验提高了我们阅读学术论文与进行分类总结的能力，了解了许多图像处理方面的新技术。也提高了我们发现问题解决问题的能力，并对程序设计有了更深入的了解。同时，全面提高了我们对 python，TensorFlow，flask，html，css，JavaScript 等知识的了解，并认识到自己的欠缺。除此之外，也让我们知道了小组合作，组内成员合理分工的重要性。从各个方面，都收获颇丰。