shelgi图像检索

这篇文章主要是想站在目前已有的文献资料、技术以及一些科技产品的基础上，结合自己的一些想法，对图像检索的方法进行整合、创新。

1. 图像检索的概述

图像检索，顾名思义就是“以图找图”，输入一个待检索的图像去数据库中进行匹配，得到最相似的图像。

根据描述图像的方式不同检索技术分为两大类：

- 基于文本的图像检索(TBIR)
- 基于内容的图像检索(CBIR)

基于文本的图像检索就是利用文本描述图像，而基于内容的图像检索注重的是图像本身的特性，例如颜色特征、纹理特征……

图像检索的主要步骤：

1. 输入图片
2. 特征提取
3. 特征矩阵之间的相似度计算
4. 返回排序结果

简单来说就是**提取特征、计算相似度**

过去往往是以传统的图像处理技术来完成第一步，例如提取图像的纹理特征、颜色特征，使用颜色矩、SIFT、LBP……

现在随着深度学习发展，更多的是以各种CNN作为特征提取器来代替传统的图像处理技术，好处在于可以直接利用在大规模数据集上训练好的预训练模型，并且有较好的性能。

一开始直接利用CNN，训练之后可以输出各类别的概率，但是不能具体到某个对应图像；然后去除最后的全连接层以及softmax层之后，就得到了输出的特征矩阵，再计算相似度就可以得到图像的相似性即实现检索。

2. 目前现有的方法

现在大多数主流的方法都是基于卷积神经网络作为特征提取器，创新点大多集中于相似度的定义、特征融合(多渠道得到特征矩阵)、利用hash进行特征编码、加入attention机制、利用GAN的变体……

其中比较厉害的就是对卷积神经网络进行的改进，因为目前分类的准确率已经很高了，从ResNet-50之后到SENet，更多的改进只是扩展已有的网络结构。而GAN在提出之后，它的变体层出不穷，特别是**DCGAN**(将CNN结合到生成器和鉴别器中)获得不错效果后，更多CNN结合GAN的模型产生，其中对检索比较有用的就是**半监督生成对抗网路(SGAN)**SGAN的优点在于不需要大数据量的样本数据而是只需要小部分带标签的数据，就能得到分类效果很好的鉴别器。（SGAN中的鉴别器不再是二分类，而是N+1分类）这也就是意味着使用鉴别器可以作为好的特征提取器。

上述都是网络层面的改进，其他的就是特征提取之后的编码、降维、相似度定义以及特征提取之前对输入图片进行数据增强等手段。

3. 主要的难点

图像检索的主要难点还是与图像本身有关，或者说与待检图像与数据库之中的图像相关性有关，主要会有以下几种情况

- 如何在数据库中的多张相似图像中找到与待检图像最相似的那一张图片
- 待检图片与原图可能拍摄角度不一致
- 待检图片与原图可能曝光率不同、亮度不一致
- 待检图片可能与原图大小尺度不一致
- 待检图片可能只是原图的一小部分边缘区域，或正好相反
- 部分遮挡

在某些不太需要注重颜色的情况下，光照的影响也许不太重要；角度问题往往可以利用图像增强（对训练图像旋转多个角度）或者利用一些旋转不变性的算子联合提取特征，做特征融合；尺度问题也可以利用图像增强(对训练图像尺度缩放)或者利用尺度不变性的算子；最难处理的就是局部区域的检索，因为大多数特征提取的结果关注点都在图像的中心而忽略了大多数边缘区域，为了解决这个问题空间验证+拓展查询可以有一定的效果。

4. 实现

对于实现一个好的图像检索，实际上就是多种方法的排列组合，尝试出一种最优的搭配。当然以创新的角度出发，还可以使用不同的方法模型。我自己有下面三种想法，针对三种不同的问题设计。

4.1 CNN+SIFT特征融合

为了解决图像的角度、光照以及尺度问题，首先对训练数据进行图像增强，输入RESNet-50进行特征提取，同时也将训练数据使用SIFT提取特征，最后进行特征融合。将融合的特征再进行VLAD编码后降维，得到最后的特征向量。

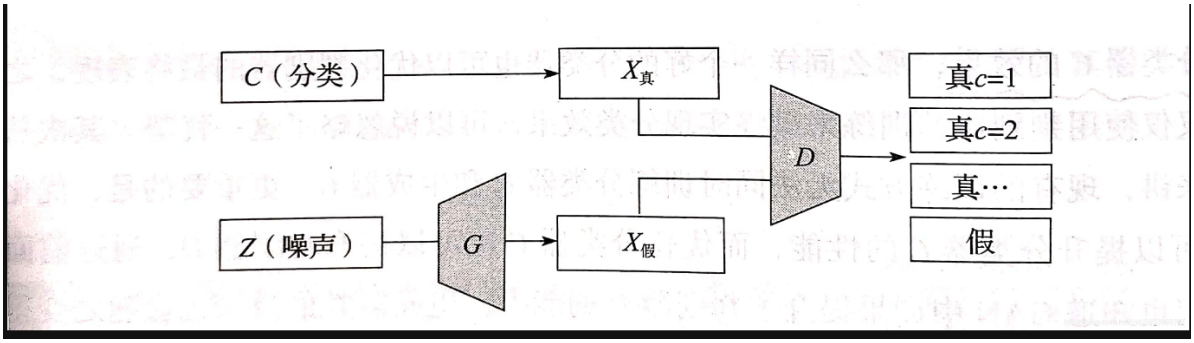
注意的地方：CNN特征提取器的提取不能取高层的特征，相反中间偏上的局部低层特征对图像检索的效果比高层特征更好，因为过于高层的特征丢失了实例特征。

缺点：这种方法的主要缺点就是训练耗时较久，检索的时候对待检图片也要进行特征融合过程，所以检索时间也是三种方法间最久的。

优点：检索的平均检索率应该是最高的，鲁棒性最强。

4.2 基于半监督生成对抗网络的图像检索

为了解决带标签训练数据集少的问题，采用半监督的生成对抗网络来实现。具体网络结构如下



通过训练，可以得到一个分类性能较好的鉴别器。往往在相同带标签数据量大小的情况下，SGAN的性能比全监督性能高20%，这里的性能主要指的是准确率。那么这种情况下提取到的图像局部特征往往也是比较合适的，再利用编码等方法进行相似度计算，就可以实现高效率检索。

优点：需要的带标签样本数据少

缺点：训练困难，往往没法得到理想的鉴别器

4.3 基于自编码器的图像检索

上面说到了GAN，那就不得不提到它的前身自编码器。GAN最大的缺点就是不稳定性，难以训练。而自编码器不同，它有固定的损失函数，目标就是重现图片，所以在编码器的时候，往往得到的就是图片的主要特征。因此，**为了解决GAN难以训练的问题**，将GAN退化为AE来作为特征提取器。

优点：训练简单易于收敛，而且检索时间较快

缺点：没有突出表现，模型性能没有第一种方法好，鲁棒性较为普通。

4.4 总结

上面我自己构想了三种方法，第一种算是面面俱到，会得到较好的平均检索率，但是会付出较大的训练成本以及检索时间成本；第二种用于标签数据量较小的情况，但是难以训练；第三种应该算是中庸之道的方法，为了时间在性能上有所妥协，但是后续改进空间较大。以上还没有添加注意力机制，因为目前我对注意力机制还不太了解，后续我进行了解后看需要再进行添加。

5. 代码部分

5.1 基于自编码器的图像检索

这个思想最为简单，而且训练成本低，所以我先尝试着做出这个demo

主要代码1

主要代码2

分别对比hash编码与最近邻查找效果

查询两次，返回最相似个数为3和5

- hash编码

```
2021-07-20 17:54:33.635404: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully
检索耗时:0.09695816040039062s
待检图片下标:20
检索图片下标: [20, 10020, 20020]
检索耗时:0.11713838577270508s
待检图片下标:20
检索图片下标: [20, 10020, 20020, 30020, 4366]
测试检索2次，查全率分别为:[1.0, 0.8]
召回率分别为:[0.75, 1.0]
平均查询检索精度:0.9

Process finished with exit code 0
```

- 最近邻查找

```
2021-07-20 17:53:05.524902: I tensorflow/stream_executor/platform/default/ds
检索耗时:0.025466442108154297s
待检图片下标:20
检索图片下标: [   20 10020 20020]
检索耗时:0.022564888000488828s
待检图片下标:20
检索图片下标: [10020 20020    20 30020 14366]
测试检索2次，查全率分别为:[1.0, 0.8]
召回率分别为:[0.75, 1.0]
平均查询检索精度:0.9
```

可以看出两种方法在性能上没有较大差异，但是在检索时间上最近邻只需要向量化计算一次全局图片距离，而hash方法则需要每一个图片比对汉明距从而得到最相似图片，因此最近邻查找耗时远远小于hash查找。而且在有限的测试集上，两种方法都得到了最优的结果（相似的图片只有四种）。

接下来的时间，我会尝试预训练模型与SIFT、HOG进行特征融合，降维之后再比对，在更大的数据集中尝试效果

5.2 基于特征融合的图像检索