# 文件存放

## 1、./ImageRetrieval/code/algorithm/目录下：

存放同源检索算法的相关代码，其中有三个.sh脚本用于运行代码：

### ①特征提取(Feature Extraction)

运行脚本feat\_extract.sh即可开始提取特征，通过修改命令中-database后的路径，可以修改需要提取特征的图像文件夹；修改-index后的文件名，可以修改生成的.h5文件的文件名。

### ②模型训练(Model Training)

运行脚本train.sh即可开始判别网络的训练。其中：

-queryindex 参数后面是query集提取的特征文件路径。

-index 参数后面是database集提取的特征文件路径。

-negativeindex 参数后面是与query集完全不同源的图像集的特征路径文件。

-model 参数后面是训练模型的保存路径。

-MaxRes 参数后面是查询时要取的前TopN中的N的取值。

-trainmode 参数为1或2，1表示从头开始训练模型，2表示载入上次训练的模型继续训练。

### ③检索查询(Query)

运行query.sh即可开始查询检索。其中：

-queryindex参数后面为query集提取的特征文件路径。

-index后面为database集提取的特征文件路径。

-model 参数后面为查询时需要调用的模型路径。

-MaxRes 参数后面为查询返回的TopN中N的数量。

-result 参数后面为输出csv文件名

## 2、./ImageRetrieval/code/data\_process/目录下：

分别存放了水印、旋转缩放、马赛克、画中画、噪声、灰度对比度6种变换的代码。

## 3、./ImageRetrieval/data/features/目录下：

存放了各个数据集提取的特征文件。featureCNN\_40.h5为40种变换图像提取的特征，featureCNN\_50.h5为50种变换图像提取的特征，featureCNN\_big.h5为60种变换图像提取的特征，featureCNN\_neg.h5为3万张与query集不同源图像提取的特征，featureCNN\_Q\_big.h5 为3k张query图像提取的特征。

## 4、./ImageRetrieval/data/images/目录下：

该目录下分为两个部分，一个名为query的文件夹存放2998张query图像；另一个名为database的文件夹存放179880张变换图像。

## 5、./ImageRetrieval/models/目录下：

该文件夹下分为三个部分，model40文件夹下存放40种变换图像进行训练得到的模型;model50文件夹下存放50种变化图像进行训练得到的模型;model60文件夹下存放60种变化图像进行训练得到的模型。

## 6、./ImageRetrieval/logs/目录下：

该文件夹下存放使用tensorflow训练时的log文件，用于tensorboard的可视化。

## 7、./ImageRetrieval/doc/目录下：

该目录下存放同源图像检索的文档。

# 模型结构、训练及算法调优

## ①模型结构

模型结构分为两个部分，第一个部分为特征提取网络，使用开源的ImageNet预训练VGG模型进行特征提取，输入一张图片，输出(512,)的向量；第二部分为判别网络，由三个全连接层组成，输入为两个长度为512的向量concatenate之后得到的长度为1024的向量，fc1神经元个数为256，fc2神经元个数为256，fc3神经元个数为2,最后fc3的输出会经过softmax层，输出两个[0,1]之间的概率。

## ②模型训练

-先选取80%的数据作为训练集，20%的数据作为测试集；

-训练采用的优化器为Adam优化器，损失函数为cross entropy loss；-训练时每个Batch将每个一维query向量扩展至120\*512，与该query向量的60种同源变换向量和60个非同源图特征相连进行拼接，得到120\*1024的张量作为网络的输入；

-训练的真值为120\*2的张量，前60\*2的张量表示正样本，即60个[0,1]向量的拼接，后60\*2的张量表示负样本，即60个[1,0]向量的拼接。

-训练过程中采用早停(early-stopping)法防止过拟合。当前epoch的平均test loss小于上一个epoch的平均test loss时即停止训练

## ③算法调优

由于模型针对画中画和旋转缩放这两种变换的召回率较低，于是考虑对这两种变换进行针对性的处理，代码位于~/ImageRetrieval /code/algorithm/SiftMatchPoint.py。算法流程如下：

1、使用SIFT算法计算query图与db图之间的匹配点个数。

2、寻找两张图片之间的Homography(单映性矩阵)即匹配点的变化矩阵，若矩阵为空，则将上一步得到的匹配点个数清零；若矩阵非空，则保留匹配点个数。

3、若匹配点个数大于10，则计算db图中匹配点的最小外包矩形，若外包矩形大小与query图尺寸不同，则保留匹配点个数；若外包矩形大小与query图尺寸一致，则将匹配点数清零。

4、最后对该query图与所有db图的匹配点个数排序，取个数最多的前20个db图，将这20张图中的矩形部分裁切出来，再进入VGG网络进行特征提取。

算法优缺点：优点是能对画中画和放缩旋转有较好的识别性，缺点时速度太慢。

# 测试结果

## ①使用40种变换(除去画中画和旋转放缩)训练出的模型在水印、马赛克、亮度对比度、噪声四种变换的图像集上测试得到的召回率为67%。

## ②使用50种变换(除去画中画)训练出的模型在水印、马赛克、亮度对比度、噪声、旋转放缩五种变换的图像集上测试得到的召回率为62%。

## ③使用60种变换训练出的模型在水印、马赛克、亮度对比度、噪声、旋转放缩、画中画六种变换的图像集上测试得到的召回率为52%。