**多媒体技术大作业1**

**2018302080152\_网安1班\_范圣悦**

1. **问题描述**

在给定的图片库里，以颜色直方图为依据，用最近邻方法或者其它方法找到某张图片的类别。随机抽取一张图片，利用模板匹配方法，在图片库中找出100幅最相似的图片，计算查全率和查准率。

1. **解题思路**
2. 利用sklearn的随机森林分类器，以颜色直方图为依据，训练数据集中的图片得到模型，利用模型预测某张图片的类别。
3. 假设源图片img真实类别为X，则在所有图片中，找到模型预测出类别为X的所有图片。
4. 为第二步找到的所有图片计算颜色直方图，利用calchist()函数找到与源图片img相似度最接近的五张照片。
5. **实验环境**

电脑配置：Intel(R) Core(TM) i5-8300H CPU @2.30GHz 2.30GHz,16G 内存

操作系统：Windows10 家庭中文版[版本 10.0.17763.1217]

开发工具：Pycharm 2020.1 x64

1. **方法流程说明**
2. **读取训练集和测试集。**

X\_train列表保存训练集图像名称，y\_train列表保存训练集图像类标；X\_test列表保存测试集图像名称，y\_test列表保存测试集图像类标；

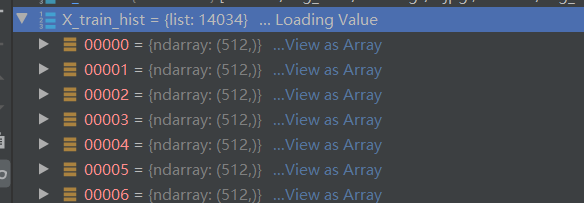
其中训练集为seg\_train文件夹，包含6个不同类别['buildings', 'forest', 'glacier', 'mountain', 'sea', 'street']的14034张图片。

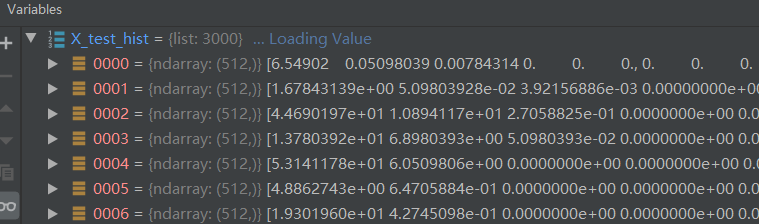
测试集为seg\_test文件夹，包含6个不同类别['buildings', 'forest', 'glacier', 'mountain', 'sea', 'street']的3000张图片。





1. 分别计算训练集和测试集的颜色直方图，经过测试发现RGB三通道直方图效果略优于HSV模型直方图，这里采用RGB三通道颜色直方图。计算后分别用X\_trian\_hist和X\_test\_hist保存得到的直方图。

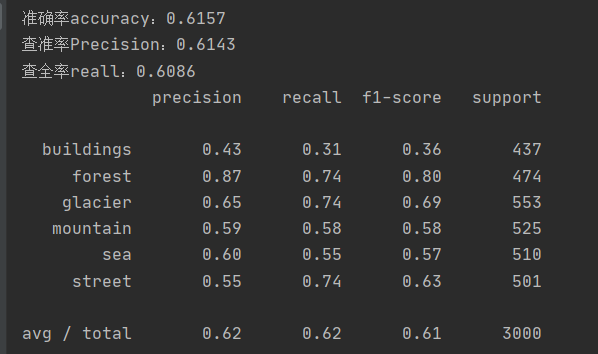




1. 使用sklearn库的随机森林分类器RandomForestClassifier，设置恰当的参数，保存模型到本地，以便下一次直接加载保存好的模型。



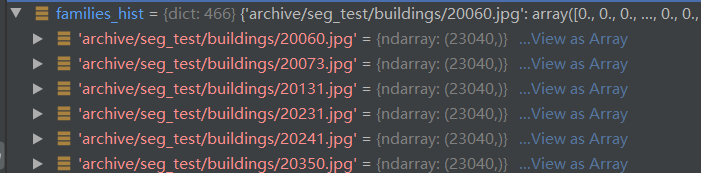
1. 利用sklearn. metrics方法计算算法的各种评价指标，分别计算其准确率accuracy、查准率precision和查全率recall，结果保留四位小数，如下图所示：



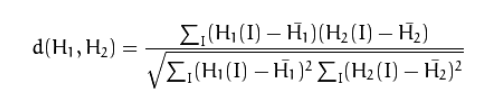
1. 找出原图片的真实类别X，利用随机森林分类器找到所有图片的预测分类与原图片真实类别X一样的图片，保存这些同类别图片的名称到列表families\_nams中。



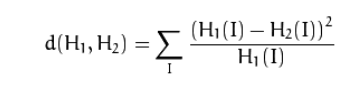
1. 计算第五步得到的所有同分类图片的颜色直方图，经过测试发现RGB三通道直方图效果差于HSV模型直方图，这里采用HSV模型颜色直方图。计算得到的直方图保存在families\_hist中。如下图所示：



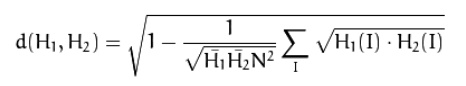
1. openCV库提供了直方图比较的方法，对输入的两张图像计算得到直方图H1与H2，归一化到相同的尺度空间，然后可以通过计算H1与H2的之间的距离得到两个直方图的相似程度，进而比较图像本身的相似程度。将同类别的所有图片颜色直方图与源图片img的颜色直方图一一比较，有4种比较方法可供选择：
   1. 相关性比较 (method=cv2.HISTCMP\_CORREL) 值越大，相关度越高，最大值为1，最小值为0



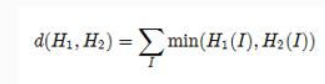
* 1. 卡方比较(method=cv2.HISTCMP\_CHISQR 值越小，相关度越高，最大值无上界，最小值0



* 1. 巴氏距离比较(method=cv2.HISTCMP\_BHATTACHARYYA) 值越小，相关度越高，最大值为1，最小值为0



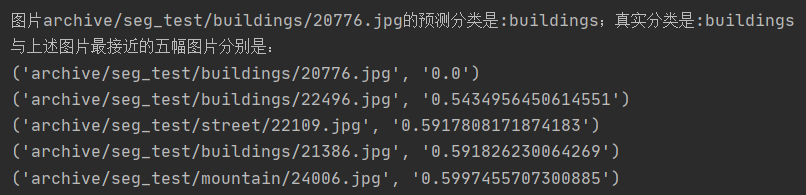
* 1. 交集法比较(method=CV\_COMP\_INTERSECT )，取两个直方图每个相同位置的值的最小值，然后求和。值越大越相似。



1. 这里选择巴氏距离比较方法，计算出所有图片与源图片的相似度后，按照从小到大的顺序排序，选择前五副作为与源图片最接近的五幅图片。

从六个不同分类中分别选择一张作为源图片，程序结果如下图所示：

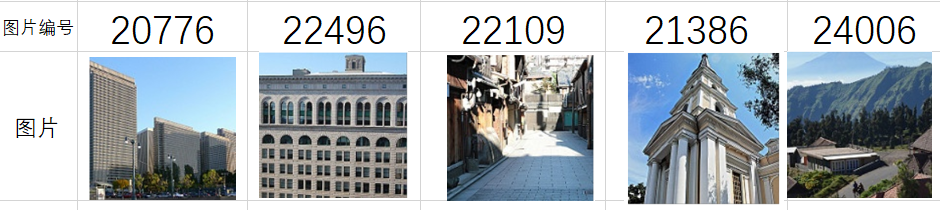
（1）buildings



源图片：

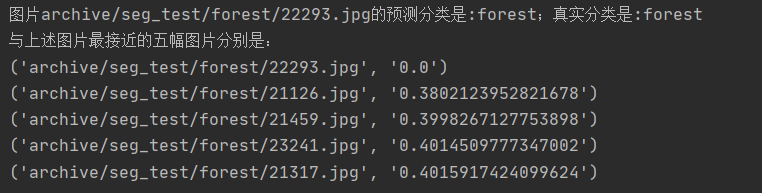


最相似的五张图片：



类别分别是buildings、buildings、street、buildings、mountain

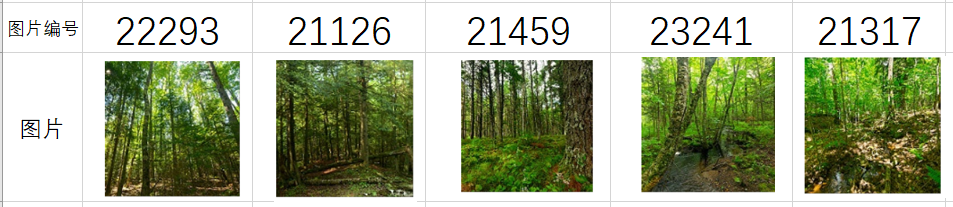
（2）forest



源图片：

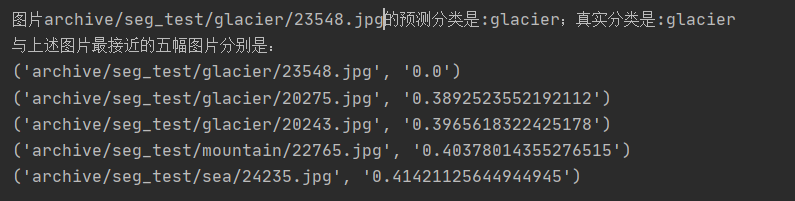


最相似的五幅图片：



类别分别是forest、forest、forest、forest、forest

（3）glacier



源图片：

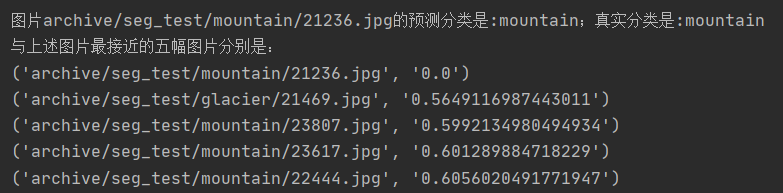


最相似的五幅图片：



类别分别是glacier、glacier、glacier、mountain、sea

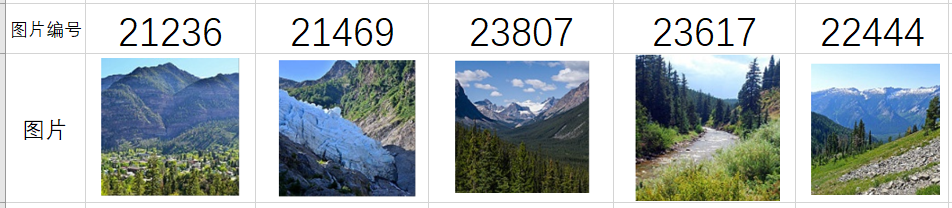
（4）mountain



源图片：

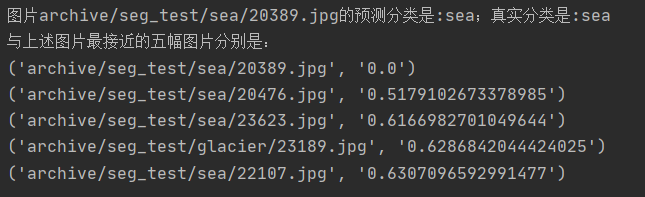


最接近的五张图片：

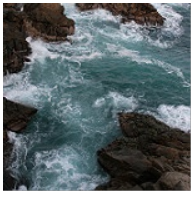


类别分别是mountain、glacier、mountain、mountain、mountain

（5）sea



源图片：

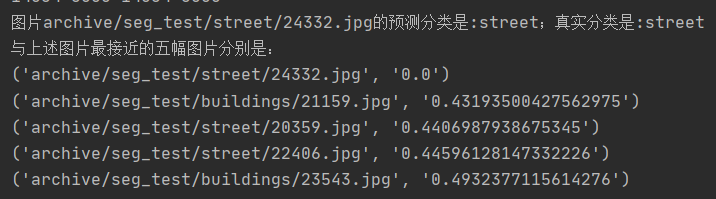


最接近的五张图片：



类别分别是：sea、sea、sea、glacier、sea

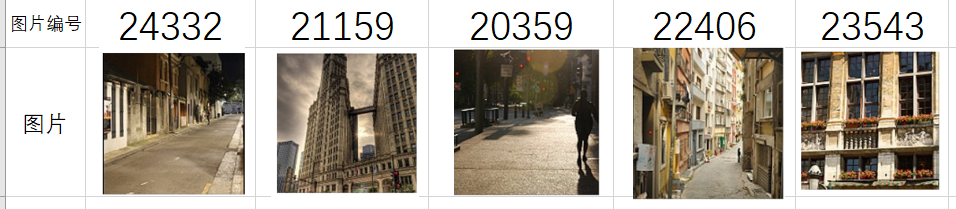
（6）street



源图片：



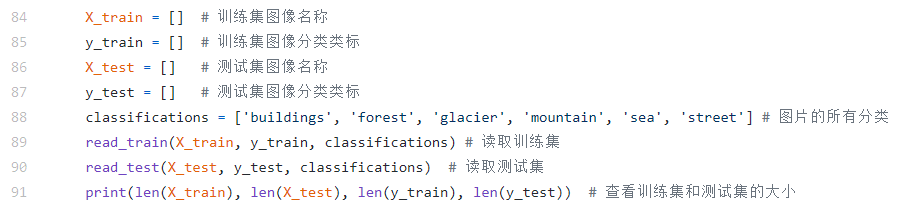
最接近的五张图片：



类别分别是：street、buildings、street、street、buildings

1. **代码与代码说明**

**1、读取训练集和测试集：**

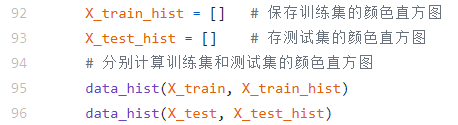
主函数调用：

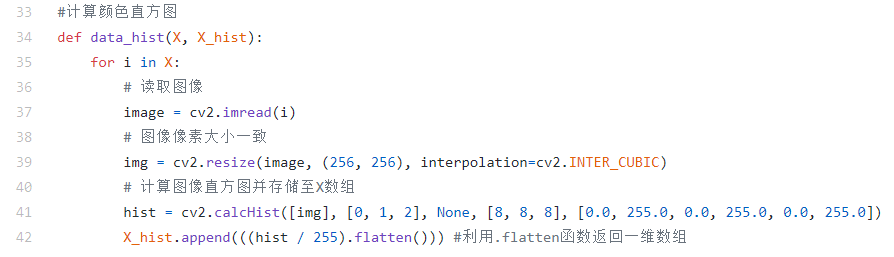
具体函数实现：

read\_train和read\_test函数分别用于读取数据集和训练集，其中X数组用于保存图片的名称（例如’archive/seg\_test/sea/20389.jpg’），Y数组保存类别标签。

**2、计算训练集和测试集的RGB颜色直方图：**

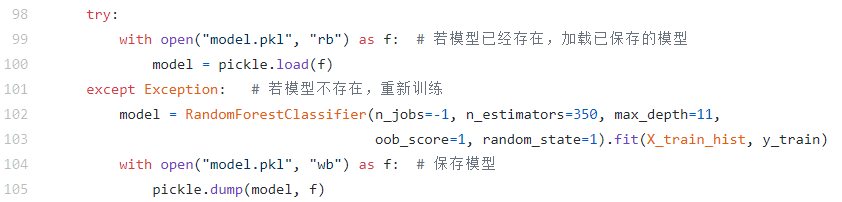
主函数调用：



具体函数实现：

data\_hist函数用于计算图片的颜色直方图，函数参数中，X代表保存图片的列表，X\_hist为计算得到的颜色直方图。首先利用cv2.resize函数将图片全部调整到同样大小，随后利用cv2.calcHist函数计算RGB三个通道颜色直方图，最后利用numpy.flatten函数得到一维数组。

利用data\_hist，先计算出测试集和训练集图片的颜色直方图，分别保存在X\_train\_hist列表和X\_test\_hist列表中。

**3、使用随机森林分类器**

首先检测模型是否已经存在，若存在，则加载已经保存的模型，否则重新训练。利用sklearn库中的RandomForestClassifier函数训练模型。

RandomForestClassifier的参数解释如下：

n\_jobs=-1：代表运用所有的处理器进行训练；

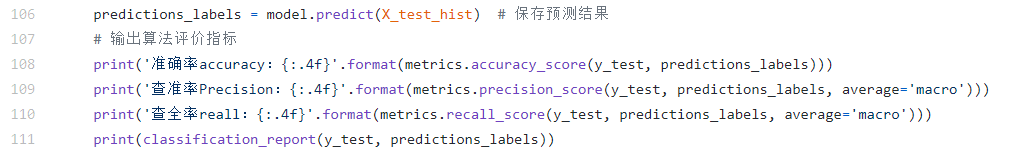
n\_estimators ：随机森林中树的数量；

max\_depth：树的最大深度；

oob\_score：使用袋外（out-of-bag）样本估计准确度；

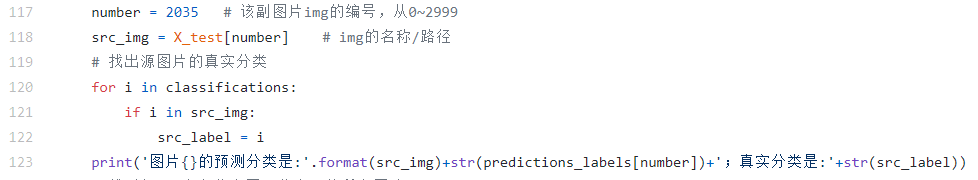
random\_state：随机数种子。

以训练集的RGB颜色直方图为输入数据，训练集的分类标签为标签对模型进行训练，训练完成后，把模型保存为model.pkl。

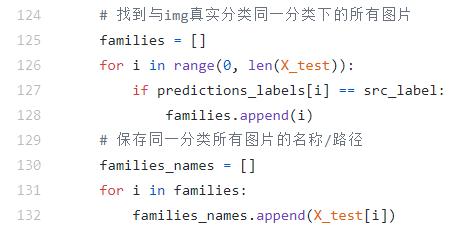
**4、对测试集进行预测和指标评价**

对测试集图片的颜色直方图进行预测，得到测试集中每一张图片的分类标签，将其保存在predictions\_labels列表中。

调用sklearn.metrics方法，分别计算准确率accuracy、查准率precision、查全率recall，均保留四位小数。最后用classification\_report函数用于显示主要分类指标的文本报告。

**5、找出将要进行对比的图片img，记录其预测类别和真实类别**

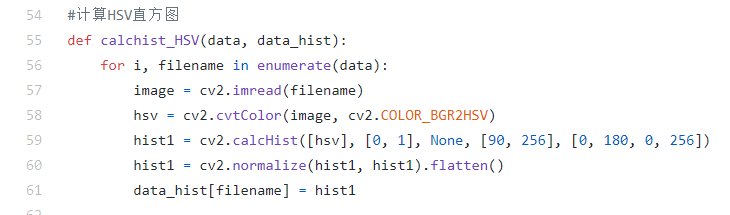
**6、找出测试集所有图片中与img同类的图片**



families列表找出与img真实分类同类别的所有图片，famalies\_names保存同一分类所有图片的名称。

**7、计算famalies\_names中所有图片的HSV颜色直方图**

主函数调用：

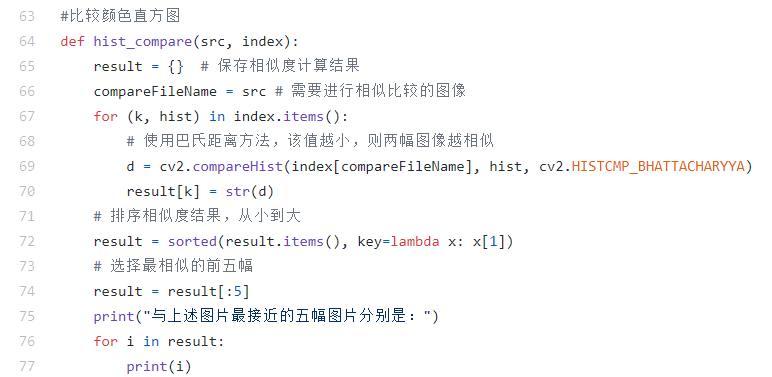
具体函数实现：

families\_hist为一个字典，每一个键表示与img同分类的图片名称，每一个键的值表示对应图片的HSV颜色直方图。

calchist\_HSV函数用于计算HSV直方图。在比较图像相似度时，HSV颜色空间的效果比RGB颜色空间更优，因此首先利用cv2.cvtColor函数转换图像的颜色空间，将RGB颜色空间转换为HSV颜色空间。随后利用calcHist函数计算颜色直方图，再将得到数据归一化处理（归一化是为了后面数据处理的方便，其次是保证程序运行时收敛加快），最后把键值添加到字典中。

**8、基于直方图进行比较，找出最接近的五幅图片**

主函数调用：

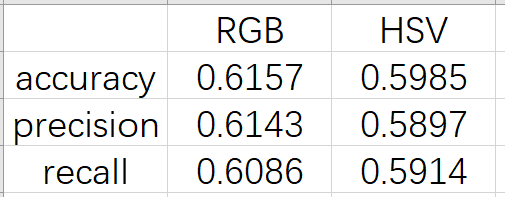
具体函数实现：

使用cv2.compareHist函数进行直方图比较，compareHist函数的第三个参数代表比较的方法，由于巴氏距离方法比较准确，所以这里采用巴氏距离。将比较后得到的相似度结果存放在result中。

把所有图片都与img比较完后，对result中的相似度结果进行排序，按照从小到大的顺序。最后截取前五副，打印输出。

1. **结论与分析**
2. **分析随机森林分类器模型的查准率、查全率优化**
   1. 不同颜色空间对模型效果的影响：

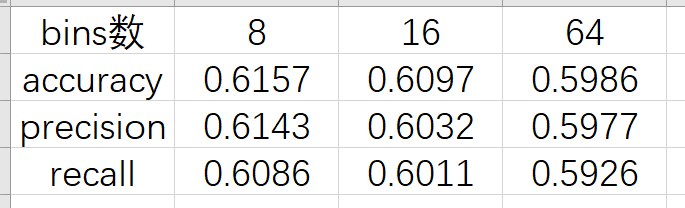
分别测试计算直方图时，选择RGB颜色空间和HSV颜色空间对模型效果的影响，得到结果如下：



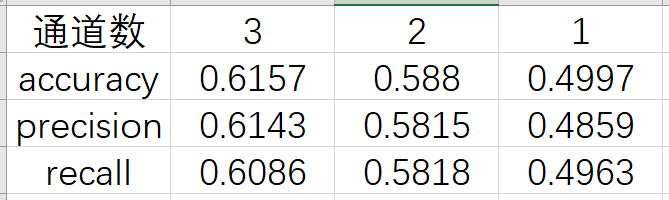
可以发现，在对数据进行特征提取的时候，选用RGB颜色空间比选用HSV的效果略好一些。

* 1. calcHist函数中参数对效果的影响：

计算RGB直方图时，calcHist中的参数histSize，表示这个直方图分成多少份（即bins，多少个直方柱）。分别测试计算直方图时，选择不同的bins数对模型效果的影响，得到结果如下（bins数太大会内存爆炸无法计算）：

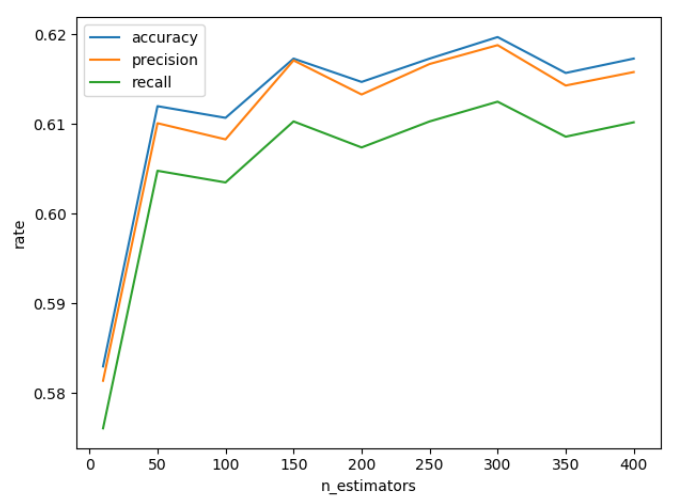


计算RGB直方图时，calcHist中的第二个参数，表示计算直方图的通道。分别测试计算直方图时，选择不同的通道数对模型效果的影响，得到结果如下：

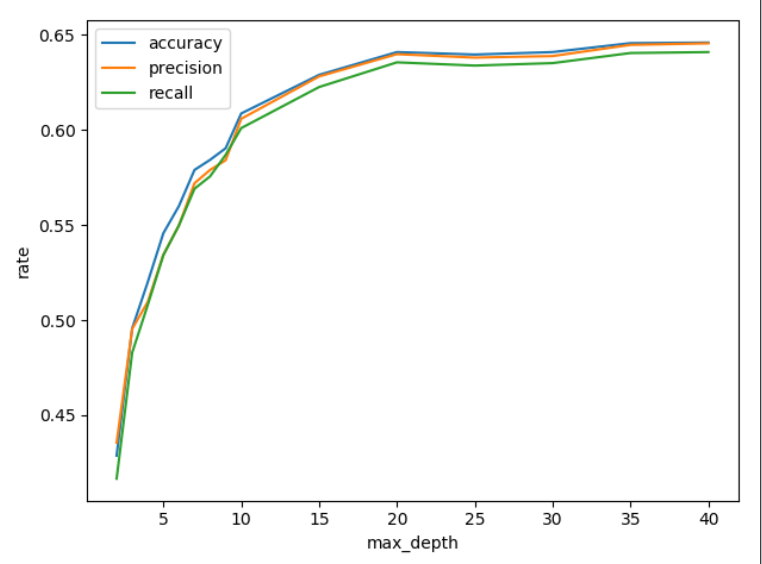


* 1. 随机森林分类器参数对效果的影响：

改变n\_estimators（随机森林中树的数量），多次测试得到改变结果如下图所示：



改变max\_depth（树的最大深度），多次测试得到改变结果如下图所示：



1. **实验结论**

由上面的参数分析可知：

* 1. 对数据进行特征提取的时候，选用RGB颜色空间比选用HSV的效果略好一些；
  2. 计算RGB直方图时，bins取8时效果最优；
  3. 随机森林分类器中树的数量在300左右时最优，但200以上变化不大；
  4. 随机森林分类器中树的深度越深，效果越好，但20以上变化不大；

同时，通过直方图比较分析得到以下结论：

* 1. 得到的最相似图片可能与原图片分类并不同。

原因是：直方图表示颜色在图像中所占的比例，主要考虑颜色，不考虑形状，无法准确描述图像内容，存在误差。

* 1. 由上述报告得到五幅最接近图片可知，他们的颜色比例都十分相近，但形状大小并不相似。
  2. 一些特定分类的图片，如海和冰川、街道和建筑，他们的颜色非常接近，因此查找海的相似图片，可能得到冰川，查找街道的相似图片，可能得到建筑；而另一些图片，如森林，由于其独特的绿色在其余分类中比较少见，通常找相似时能够找到同一类别。

本次项目的GitHub地址：https://github.com/fshooty379/2020multi-media