

**模式识别大作业**

题 目 叶子分类

学 院 信息科学与工程

专 业 信息与通信工程

组 员 何超逸

指导教师 赵海涛

**完成日期： 2019 年 12 月1日**

# 模式识别作业报告**——叶子识别**

## **叶子识别简介**

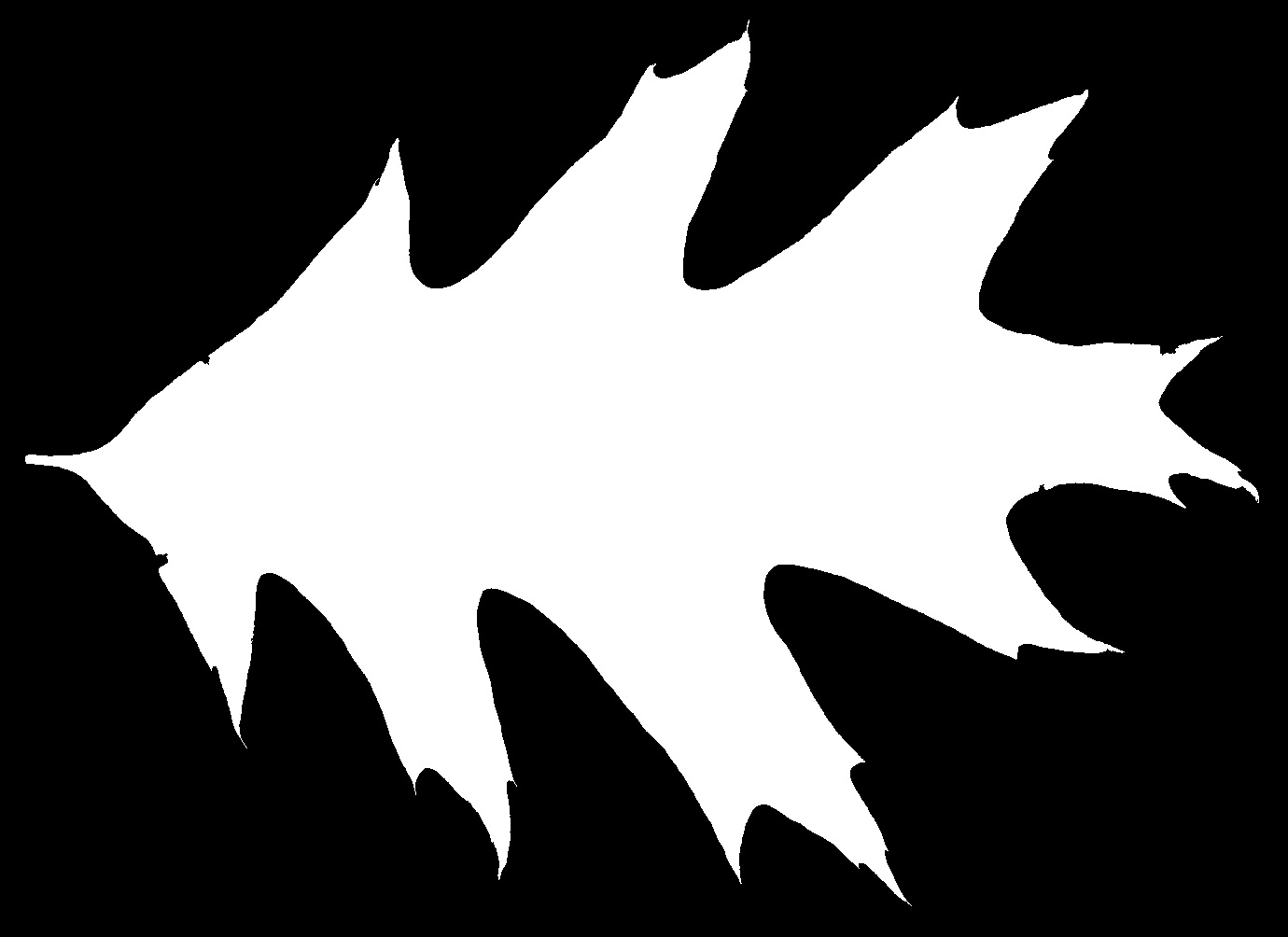
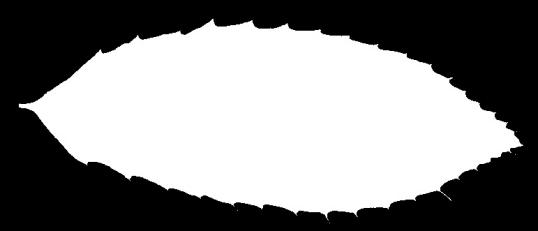
据估计，世界上大约有50万种植物。物种分类在历史上一直存在问题，常常导致重复的鉴定。叶片因其体积大、分布广、特性独特，是鉴别植物种类的有效手段。

## 整体解决方案

这是一个分类问题，可以用一些分类算法来实现。

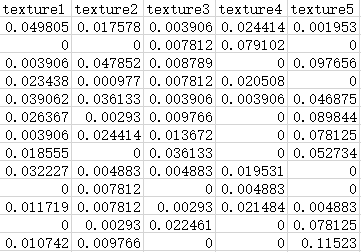
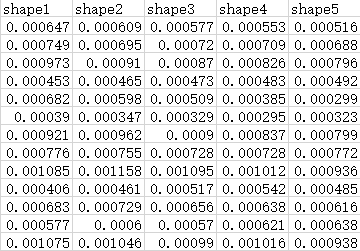
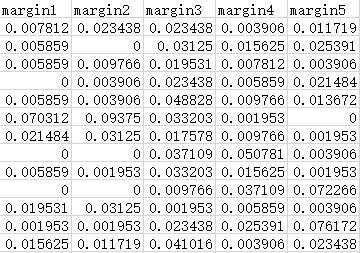
### 2.1数据结构分析

该数据集包括大约1584个图像的叶片标本（16个样品，每99种），已转化为二进制黑色叶片对白色背景。如图1所示。



**图1 样本图示**

每个图像还提供三组特征：形状连续描述符、内部纹理直方图和一个精确的边缘直方图。对于每个特征，每个叶样本都有一个64属性向量。如图2。



**图2 数据集**

这里只截取了每个特征的前5个属性向量，初始的数据已经经过了量化，由于参数都比较小，空白的位置也做了填0的处理。所以这组初始的数据集做分类还是比较方便的。

训练集和测试集分别为991\*194，595\*100的csv文件。共有1584个样本（99种树叶各16个样本）。

### 2.2 数据导入

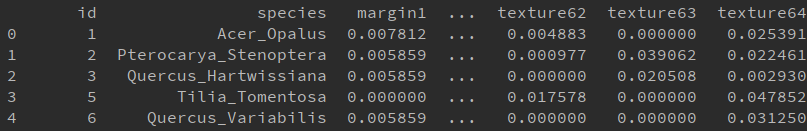
|  |
| --- |
| train = pd.read\_csv(‘train.csv’)  test = pd.read\_csv(‘test.csv’) |

使用pandas读入即可。

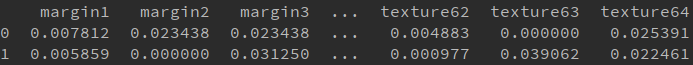
### 2.3 数据预处理

对数据进行标签编码，将species和id保存到classes和test\_ids中后，移除这两列。

|  |
| --- |
| def encode(train, test):  le = LabelEncoder().fit(train.species)  labels = le.transform(train.species)  classes = list(le.classes\_)  test\_ids = test.id    train = train.drop(['species', 'id'], axis=1)  test = test.drop(['id'], axis=1)  return train, labels, test, test\_ids, classes |



**图3 初始数据集**



**图4 处理后的数据集**

图3、图4分别为原始的数据集和处理后的数据集。

### 2.4 训练集划分

为了评估出哪种分类器效果比较好，在这里训练集和测试集比例为4：1。

|  |
| --- |
| StratifiedShuffleSplit(labels, 10, test\_size=0.2, random\_state=23) |

### 2.5 机器学习模型选择

|  |
| --- |
| from sklearn.neighbors import KNeighborsClassifier  from sklearn.tree import DecisionTreeClassfier  from sklearn.ensemble import RandomForestClassifier  classifiers = [KNeighborsClassifier(3),  DecisionTreeClassifier(),  RandomForestClassifier()] |

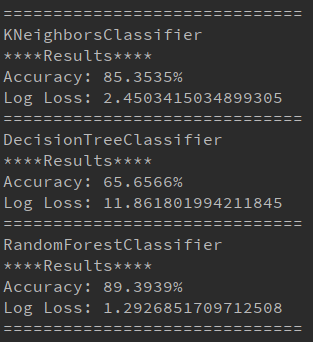
使用sklearn里的k近邻分类器，决策树分类器，随机森林分类器。

### 2.6 结果展示

|  |
| --- |
| from sklearn.metrics import accuracy\_score, log\_loss |

依然使用sklearn里的accuracy，loss

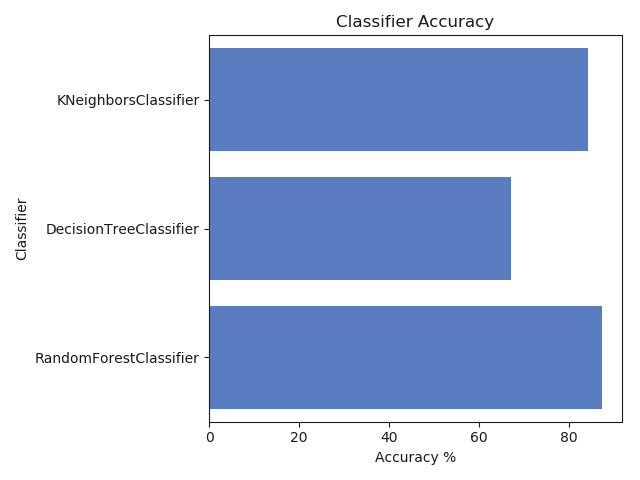
输出的结果如图5所示。



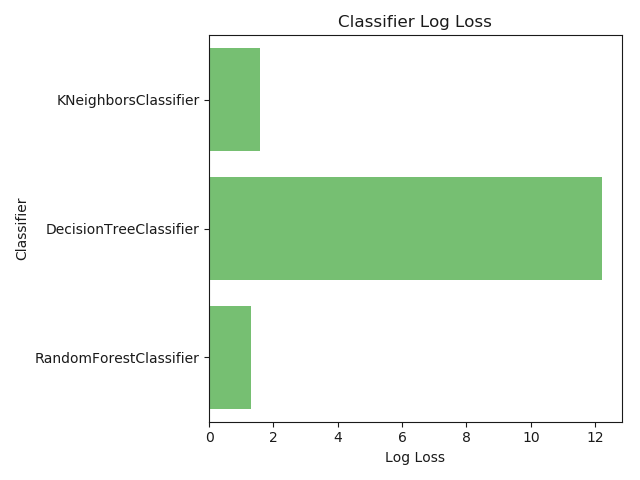
**图5 结果显示**

可见随机森林和k近邻的精度都达到了80多，且loss很低。而决策树最终的分类效果就要差一点。

图6、图7，可视化。



**图6 accuracy**



**图7 loss**

## 作业总结

经过半个学期的模式识别课程的学习，在赵老师的带领下，我逐渐对机器学习这个领域有了一点了解，并且在上课和复习的过程中，对算法的推导能力有了提升。

在这次的大作业中，由于自己的能力确实需要不断的提高，最终只是勉强实现，对于其中的问题并不能有效的解决（在某些算法的具体实现上确实有困难，对于性能的提升也不尽了解）。

但课程的学习还是收获颇丰，有风趣幽默的老师讲课是一件很幸福的事，并且锻炼了自己公式得推导能力和思维逻辑能力。由于对这个领域确实不甚了解，在接下来的研究生生涯中，如果有机会再接触这个领域的学习，我一定会运用这个课程中学到的东西，并不断的提高自己。