

# practice2 特征工程记录

对不同预处理方式的研究

original

处理过程：Equalize+Image\_Split+SIFT+HOG+LBP+4C-SVC+RandomForestClassifier+ExtraTreesClassifier+XGBooster

源码运行结果记录如下：

```
D:\AI\envs\zpython\python.exe D:\Machine_Learn\practice\task2\main.py
[ WARN:003.777] global loadsave.cpp:244 cv::findDecoder imread_('data/train/Charlock/0edcd02cd.png'): can't open/read file: check file path/integrity
train文件夹文件读取中.....
100%|██████████| 4440/4440 [00:20<00:00, 216.08it/s]
100%|██████████| 4440/4440 [00:15<00:00, 282.04it/s]
0%|          | 0/1104 [00:00<?, ?it/s]test文件夹文件读取中.....
100%|██████████| 1104/1104 [00:05<00:00, 194.26it/s]
数据加载完成，train图片数量为：4440
train_dataset: (4440, 256, 256, 3),train_label: (4440,)
test_dataset: (1104, 256, 256, 3)
0%|          | 0/5544 [00:00<?, ?it/s]1、开始进行所有图片的sift特征运算
100%|██████████| 5544/5544 [00:46<00:00, 119.26it/s]
*发现93张图片未计算出sift特征*
2.1、开始堆叠所有sift特征
shape变化: (5544, x, 128) -> (992293, 128)
2.2、开始K聚类：聚类中心个数80，数据点个数992293（这个过程耗时最长）
0%|          | 0/5544 [00:00<?, ?it/s]
下面开始HOG特征提取
100%|██████████| 5544/5544 [01:51<00:00, 49.82it/s]

PCA降维中.....
48600降至60维
0%|          | 0/5544 [00:00<?, ?it/s]
下面开始LBP特征提取
100%|██████████| 5544/5544 [00:51<00:00, 107.30it/s]

PCA降维中.....
256降至80维
```






反复进行了5次

```

=====
选用分类器: SVC(C=12, probability=True)
选用分类器: RandomForestClassifier(bootstrap=False, max_depth=40, n_estimators=400,
                                   random_state=10)
选用分类器: ExtraTreesClassifier(max_depth=50, n_estimators=400)
选用分类器: XGBClassifier(base_score=None, booster=None, callbacks=None,
                           colsample_bylevel=None, colsample_bynode=None,
                           colsample_bytree=None, device=None, early_stopping_rounds=None,
                           enable_categorical=False, eval_metric='mlogloss',
                           feature_types=None, gamma=None, grow_policy=None,
                           importance_type=None, interaction_constraints=None,
                           learning_rate=0.13, max_bin=None, max_cat_threshold=None,
                           max_cat_to_onehot=None, max_delta_step=None, max_depth=5,
                           max_leaves=None, min_child_weight=None, missing=nan,
                           monotone_constraints=None, multi_strategy=None, n_estimators=300,
                           n_jobs=None, nthread=10, num_parallel_tree=None, ...)
正确个数: 758, 总数: 888, 正确率0.853604
m_precision: 0.8501392419583023
m_recall: 0.8149792670611121
f1-score: 0.8246520463096254
=====

```

K折交叉验证次数为5次，选择了四个分类器，最终输出的csv文件中，预测的标签选择来自四个分类器中预测概率最大一个。

 submission_features0.85641740829122.csv	2023-11-06 11:13	XLS 工作表
 submission_features0.874750963217986.csv	2023-11-06 11:14	XLS 工作表
 submission_features0.8394301653901198.csv	2023-11-06 11:12	XLS 工作表
 submission_features0.8501392419583023.csv	2023-11-06 11:11	XLS 工作表
 submission_features0.8558789098643699.csv	2023-11-06 11:15	XLS 工作表

11-06-1

处理过程：Equalize+sharpening（拉普拉斯算子）

+Image\_Split+SIFT+HOG+LBP+4C-

SVC+RandomForestClassifier+ExtraTreesClassifier+XGBooster

```

D:\AI\envs\zxytorch\python.exe D:\Machine_Learn\practice\task2\main.py
15it [00:53, 3.55s/it]
 0%|          | 0/4440 [00:00<?, ?it/s]train文件夹文件读取中.....
100%|██████████| 4440/4440 [00:15<00:00, 279.75it/s]
100%|██████████| 4440/4440 [00:15<00:00, 290.21it/s]
 0%|          | 0/1104 [00:00<?, ?it/s]test文件夹文件读取中.....
100%|██████████| 1104/1104 [00:05<00:00, 211.22it/s]
数据加载完成, train图片数量为: 4440
train_dataset: (4440, 256, 256, 3), train_label: (4440,)
test_dataset: (1104, 256, 256, 3)
 0%|          | 0/5544 [00:00<?, ?it/s]1、开始进行所有图片的sift特征运算
100%|██████████| 5544/5544 [00:46<00:00, 119.62it/s]
*发现96张图片未计算出sift特征*
2.1、开始堆叠所有sift特征
shape变化: (5544, x, 128) -> (1165586, 128)
2.2、开始K聚类: 聚类中心个数80, 数据点个数1165586 (这个过程耗时最长)






下面开始HOG特征提取
100%|██████████| 5544/5544 [01:48<00:00, 50.99it/s]

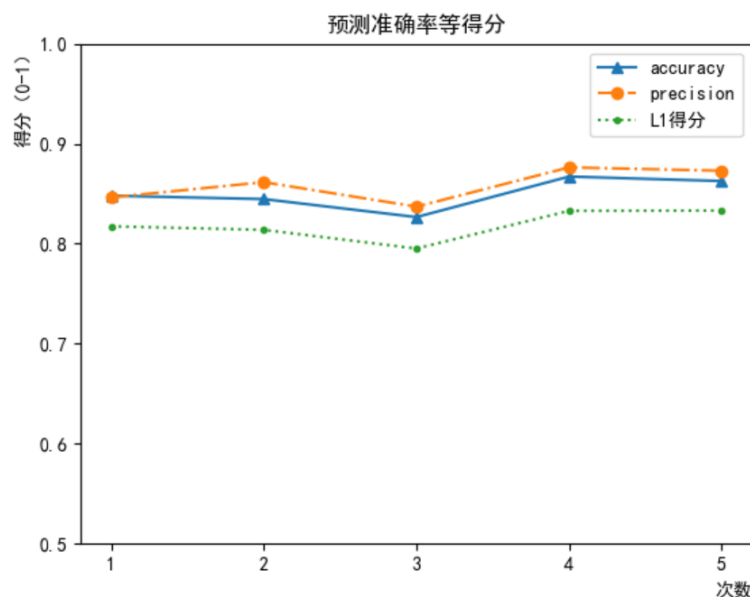
PCA降维中.....
48600降至60维

下面开始LBP特征提取
100%|██████████| 5544/5544 [00:52<00:00, 106.56it/s]

PCA降维中.....
256降至80维
=====

```

	submission_features0.86151674964822.csv	2023-11-06 14:35	XLS 工作表	34 KB
	submission_features0.8370013480125319.c...	2023-11-06 14:36	XLS 工作表	34 KB
	submission_features0.8462767140221726.c...	2023-11-06 14:34	XLS 工作表	34 KB
	submission_features0.8729176676294615.c...	2023-11-06 14:38	XLS 工作表	34 KB
	submission_features0.8762914700640602.c...	2023-11-06 14:37	XLS 工作表	34 KB



调整了颜色区间范围

绿色下限green\_lower = [35,43,46]

绿色下限green\_lower = [29,43,46]






绿色上限green\_upper = [77,255,255]

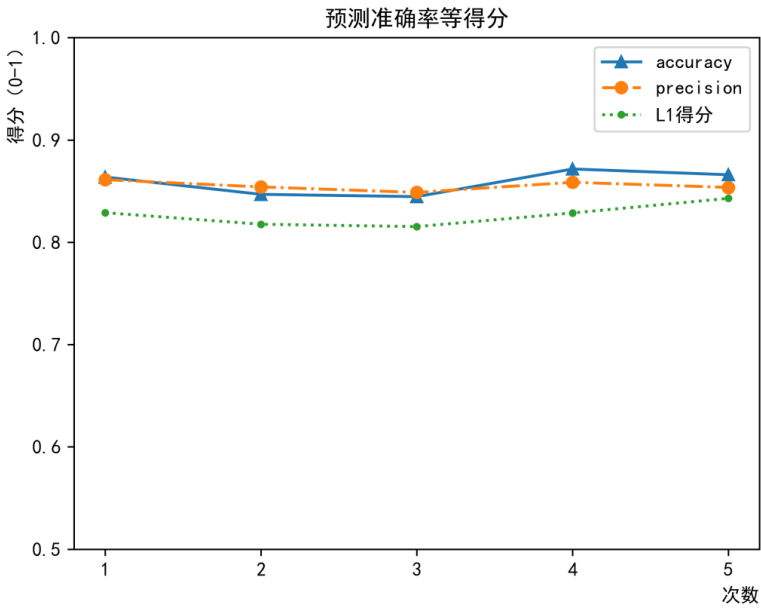
绿色上限green\_upper = [77,255,255]

```
D:\AI\envs\zxp\python.exe D:\Machine_Learn\practice\task2\main.py
train文件文件夹读取中....
100%|██████████| 4440/4440 [00:15<00:00, 280.53it/s]
100%|██████████| 4440/4440 [00:15<00:00, 294.00it/s]
0%|          | 0/1104 [00:00<?, ?it/s]test文件文件夹读取中....
100%|██████████| 1104/1104 [00:04<00:00, 225.77it/s]
数据加载完成, train图片数量为: 4440
train_dataset: (4440, 256, 256, 3), train_label: (4440,)
test_dataset: (1104, 256, 256, 3)
0%|          | 0/5544 [00:00<?, ?it/s]1. 开始进行所有图片的sift特征运算
100%|██████████| 5544/5544 [00:44<00:00, 123.37it/s]
*发现3张图片未计算由sift特征*
2.1. 开始堆叠所有sift特征
shape变化: (5544, x, 128) -> (1288692, 128)
2.2. 开始K聚类: 聚类中心个数80, 数据点个数1288692 (这个过程耗时最长)
0%|          | 0/5544 [00:00<?, ?it/s]
下面开始HOG特征提取
100%|██████████| 5544/5544 [01:46<00:00, 52.06it/s]

PCA降维中....
48600降至60维
0%|          | 0/5544 [00:00<?, ?it/s]
下面开始LBP特征提取
100%|██████████| 5544/5544 [00:51<00:00, 108.30it/s]

PCA降维中....
256降至80维
=====
```

	submission_features0.858627097019972.csv	2023-11-06 15:15	XLS 工作表	34 KB
	submission_features0.8487289133350756.c...	2023-11-06 15:14	XLS 工作表	34 KB
	submission_features0.8536185409103182.c...	2023-11-06 15:16	XLS 工作表	34 KB
	submission_features0.8540901547178095.c...	2023-11-06 15:13	XLS 工作表	34 KB
	submission_features0.8611517531022814.c...	2023-11-06 15:12	XLS 工作表	34 KB



增大了颜色区间范围后，正确率下降了，不太清楚原因

这个地方又利用小批量进行了实验，如果不考虑数据量过小带来的影响，增大颜色区间后平均正确率确实下降了，因此最终数据预处理部分不做额外变化。

```
train文件夹文件读取中.....
100%|██████████| 600/600 [00:02<00:00, 276.90it/s]
100%|██████████| 600/600 [00:02<00:00, 287.18it/s]
0%|          | 0/50 [00:00<?, ?it/s]test文件夹文件读取中.....
100%|██████████| 50/50 [00:00<00:00, 207.42it/s]
数据加载完成. train图片数量为: 600
train_dataset: (600, 256, 256, 3),train_label: (600,)
test_dataset: (50, 256, 256, 3)
1、开始进行所有图片的sift特征运算
100%|██████████| 650/650 [00:05<00:00, 119.65it/s]
*发现9张图片未计算出sift特征*
2.1、开始堆叠所有sift特征
shape变化: (650, x, 128) -> (140462, 128)
2.2、开始K聚类: 聚类中心个数80, 数据点个数140462 (这个过程耗时最长)
选用分类器: SVC(C=12, probability=True)
正确个数: 83,总数: 120, 正确率0.691667
```

```
train文件夹文件读取中.....
100%|██████████| 600/600 [00:02<00:00, 275.65it/s]
100%|██████████| 600/600 [00:02<00:00, 290.67it/s]
0%|          | 0/50 [00:00<?, ?it/s]test文件夹文件读取中.....
100%|██████████| 50/50 [00:00<00:00, 211.32it/s]
数据加载完成. train图片数量为: 600
train_dataset: (600, 256, 256, 3),train_label: (600,)
test_dataset: (50, 256, 256, 3)
1、开始进行所有图片的sift特征运算
100%|██████████| 650/650 [00:05<00:00, 122.97it/s]
*发现9张图片未计算出sift特征*
2.1、开始堆叠所有sift特征
shape变化: (650, x, 128) -> (140462, 128)
2.2、开始K聚类: 聚类中心个数80, 数据点个数140462 (这个过程耗时最长)
选用分类器: SVC(C=12, probability=True)
正确个数: 84,总数: 120, 正确率0.700000
```

为了加快探究实验的进行，按照样例报告中的设置采用小批量实验

所以在探究实验中，使用以下设置缩短单次实验所需时间：

1. 使用的train训练集中，每一类物种的图片数量都限制为50张（12类，共600张图片），位于“small\_data”文件夹下；
2. SIFT+词袋模型提取出的特征用于训练分类器时的准确率是三种中单独使用时最高的，对模型整体准确率的提高作用最为显著，所以单独使用SIFT特征进行探究实验，以减少程序一次运行所需时间；
3. 仅使用SVC（在相同特征提取的前提下，该方法正确率最高）一种分类器进行训练，以减少程序一次运行所需时间。

为了对分类器进行选择，我们测试了在相同SIFT+词袋模型提取特征下不同分类器的表现：

这个SIFT是对所有样本都处理了的，来自于all\_data，其它部分与上面保持一致

sift\_features\_bow=80.pkl

SVC如下：

选用分类器： `SVC(C=12, probability=True)`  
正确个数： 711, 总数： 888, 正确率0.800676

随机森林如下：

选用分类器： `RandomForestClassifier(bootstrap=False, max_depth=40, n_estimators=400, random_state=10)`  
正确个数： 677, 总数： 888, 正确率0.762387

极度随机森林如下：

选用分类器： `ExtraTreesClassifier(max_depth=50, n_estimators=400)`  
正确个数： 663, 总数： 888, 正确率0.746622

XGBC如下：

选用分类器： `XGBClassifier(base_score=None, booster=None, callbacks=None, colsample_bylevel=None, colsample_bynode=None, colsample_bytree=None, device=None, early_stopping_rounds=None, enable_categorical=False, eval_metric='mlogloss', feature_types=None, gamma=None, grow_policy=None, importance_type=None, interaction_constraints=None, learning_rate=0.13, max_bin=None, max_cat_threshold=None, max_cat_to_onehot=None, max_delta_step=None, max_depth=5, max_leaves=None, min_child_weight=None, missing=nan, monotone_constraints=None, multi_strategy=None, n_estimators=300, n_jobs=None, nthread=10, num_parallel_tree=None, ...)`  
正确个数： 699, 总数： 888, 正确率0.787162

K近邻算法如下：

选用分类器： `KNeighborsClassifier()`  
正确个数： 616, 总数： 888, 正确率0.693694

接下来通过小样本对预处理部分进行实验

条件：小规模样本50+均衡化+锐化+分割+SIFT+词袋模型（80）+SVC

## 锐化方法：Emboss滤波器

train文件夹文件读取中.....

100%|██████████| 600/600 [00:02<00:00, 269.95it/s]

100%|██████████| 600/600 [00:02<00:00, 285.90it/s]

test文件夹文件读取中.....

100%|██████████| 50/50 [00:00<00:00, 207.42it/s]

数据加载完成, train图片数量为: 600

train\_dataset: (600, 256, 256, 3), train\_label: (600,)

test\_dataset: (50, 256, 256, 3)

1、开始进行所有图片的sift特征运算

100%|██████████| 650/650 [00:05<00:00, 116.39it/s]

\*发现0张图片未计算出sift特征\*

2.1、开始堆叠所有sift特征

shape变化: (650, x, 128) -> (216481, 128)

2.2、开始K聚类: 聚类中心个数80, 数据点个数216481 (这个过程耗时最长)

选用分类器: SVC(C=12, probability=True)

正确个数: 77, 总数: 120, 正确率0.641667

## 锐化方法：拉普拉斯算子 该算子表现优于Emboss滤波器

train文件夹文件读取中.....

100%|██████████| 600/600 [00:02<00:00, 279.43it/s]

100%|██████████| 600/600 [00:02<00:00, 287.54it/s]

0%|██████████| 0/50 [00:00<?, ?it/s]test文件夹文件读取中.....

100%|██████████| 50/50 [00:00<00:00, 210.04it/s]

数据加载完成, train图片数量为: 600

train\_dataset: (600, 256, 256, 3), train\_label: (600,)

test\_dataset: (50, 256, 256, 3)

1、开始进行所有图片的sift特征运算

100%|██████████| 650/650 [00:05<00:00, 121.16it/s]

\*发现0张图片未计算出sift特征\*

2.1、开始堆叠所有sift特征

shape变化: (650, x, 128) -> (156058, 128)

2.2、开始K聚类: 聚类中心个数80, 数据点个数156058 (这个过程耗时最长)

选用分类器: SVC(C=12, probability=True)

正确个数: 82, 总数: 120, 正确率0.683333

train文件夹文件读取中.....

100%|██████████| 600/600 [00:02<00:00, 265.63it/s]

100%|██████████| 600/600 [00:02<00:00, 278.11it/s]

test文件夹文件读取中.....

100%|██████████| 50/50 [00:00<00:00, 203.97it/s]

数据加载完成, train图片数量为: 600

train\_dataset: (600, 256, 256, 3), train\_label: (600,)

test\_dataset: (50, 256, 256, 3)

1、开始进行所有图片的sift特征运算

100%|██████████| 650/650 [00:05<00:00, 119.63it/s]

\*发现0张图片未计算出sift特征\*

2.1、开始堆叠所有sift特征

shape变化: (650, x, 128) -> (156058, 128)

2.2、开始K聚类: 聚类中心个数80, 数据点个数156058 (这个过程耗时最长)

选用分类器: SVC(C=12, probability=True)

正确个数: 86, 总数: 120, 正确率0.716667

## 锐化方法：Sobel-Feldman算子 水平边缘检测

```
D:\AI\envs\zxpytorch\python.exe D:\Machine_Learn\practice\task2\SIFT.py
```

```
train文件夹文件读取中.....
```

```
100%|██████████| 600/600 [00:02<00:00, 271.76it/s]
```

```
100%|██████████| 600/600 [00:02<00:00, 284.46it/s]
```

```
test文件夹文件读取中.....
```

```
100%|██████████| 50/50 [00:00<00:00, 208.61it/s]
```

```
数据加载完成, train图片数量为: 600
```

```
train_dataset: (600, 256, 256, 3), train_label: (600,)
```

```
test_dataset: (50, 256, 256, 3)
```

```
0%|          | 0/650 [00:00<?, ?it/s]1、开始进行所有图片的sift特征运算
```

```
100%|██████████| 650/650 [00:05<00:00, 119.67it/s]
```

```
*发现0张图片未计算出sift特征*
```

```
2.1、开始堆叠所有sift特征
```

```
shape变化: (650, x, 128) -> (166871, 128)
```

```
2.2、开始K聚类: 聚类中心个数80, 数据点个数166871 (这个过程耗时最长)
```

```
选用分类器: SVC(C=12, probability=True)
```

```
正确个数: 73, 总数: 120, 正确率0.608333
```

发现不同算子会影响sift特征的个数, 而且由于初始化的影响, SVC每次结果不相同, 综合来看, 我们仍然使用拉普拉斯算子。

补充: 小规模测试中, SIFT\_BOW\_size的小范围变化对最终正确率影响非常小

对不同特征提取方法的研究, 经过查阅相关资料与原报告种相关部分的说明, 没有做其它的修改和补充











对于分类器集成方法的研究, 计划通过StackingClassifier的方法来进行, 相较于原方法, 该方法进一步将各分类器的输出作为输入, 学习最终的输出 (方法包括线性组合等, 取决于元分类器的设定), 从而发挥不同分类器的表现。

使用以下设置进行训练学习, 上面四个分类器各参数设置完全按照原设定。



2 usages

```
def Stacking(data, label):
    base_classifiers = [
        ('rf', RandomForestClassifier(n_estimators=400, max_features='sqrt', max_depth=40,
                                     bootstrap=False, oob_score=False, random_state=10)),
        ('SVC', SVC(C=12, probability=True)),
        ('Et', ExtraTreesClassifier(n_estimators=400, max_features='sqrt', max_depth=50)),
        ('XG', XGBClassifier(learning_rate=0.13, max_depth=5, n_estimators=300, nthread=10,
                             use_label_encoder=False, eval_metric='mlogloss'))
    ]
    meta_classifier = LogisticRegression()
    stacking_classifier = StackingClassifier(estimators=base_classifiers, final_estimator=meta_classifier)
    stacking_classifier.fit(data, label) # 训练堆叠分类器
    return stacking_classifier # 进行预测
```

 submission_features0.8512180523692755.c...	2023-11-07 15:13	XLS 工作表	2 KB	 c-submission_features0.850700980406067...	2023-11-07 15:20	XLS 工作表	2 KB
 submission_features0.8519409275995317.c...	2023-11-07 15:14	XLS 工作表	2 KB	 c-submission_features0.853744809940298...	2023-11-07 15:24	XLS 工作表	2 KB
 submission_features0.8522315344006796.c...	2023-11-07 15:15	XLS 工作表	2 KB	 c-submission_features0.857862620852666...	2023-11-07 15:39	XLS 工作表	2 KB
 submission_features0.8579014004755309.c...	2023-11-07 15:12	XLS 工作表	2 KB	 c-submission_features0.859578316617603...	2023-11-07 15:34	XLS 工作表	2 KB
 submission_features0.8611794659651525.c...	2023-11-07 15:11	XLS 工作表	2 KB	 c-submission_features0.873877487052607...	2023-11-07 15:29	XLS 工作表	2 KB

=====采用分类器学习进行分类器集成=====

正确个数: 771, 总数: 888, 正确率0.868243

m\_precision: 0.8507009804060671

m\_recall: 0.8407343636548569

f1-score: 0.8430710713387314

=====采用分类器学习进行分类器集成=====

正确个数: 766, 总数: 888, 正确率0.862613

m\_precision: 0.8537448099402987

m\_recall: 0.8501360483650019

f1-score: 0.8486955505084763

=====采用分类器学习进行分类器集成=====

正确个数: 774, 总数: 888, 正确率0.871622

m\_precision: 0.8738774870526078

m\_recall: 0.8506167107441067

f1-score: 0.8558696917667779

=====采用分类器学习进行分类器集成=====

正确个数: 781, 总数: 888, 正确率0.879505

m\_precision: 0.8595783166176036

m\_recall: 0.8477055950254825

f1-score: 0.8475856316664273

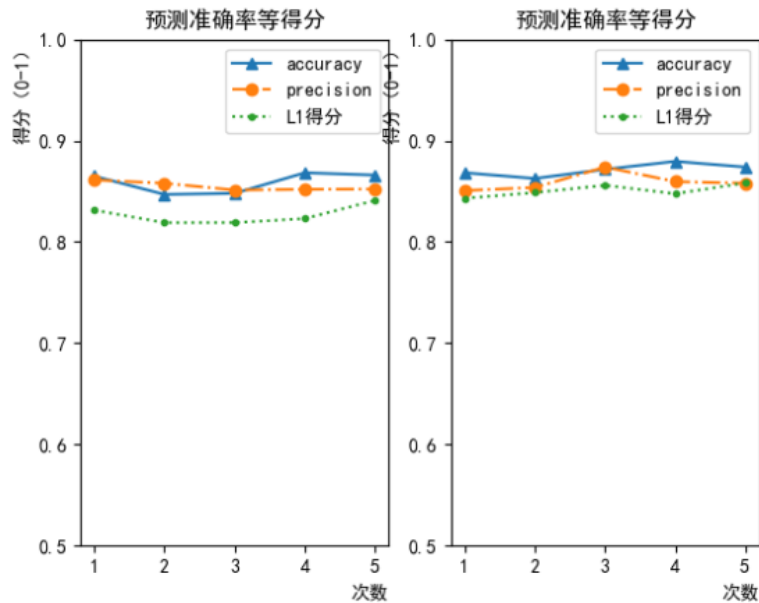
=====采用分类器学习进行分类器集成=====

正确个数: 776, 总数: 888, 正确率0.873874

m\_precision: 0.8578626208526661

m\_recall: 0.8604798277981814

f1-score: 0.8581750353129182



采用新的集成学习方法之后，平均正确率与F1值有了明显的上升，目前存在一个问题就是不太理解精确率，正确率，召回率，F1这些不同指标之间变化的关系。但该方法训练时间较长。

小规模：计划接下来采用新的集成学习方式来进行。











```
def Stacking(data, label):
    base_classifiers = [
        ('rf', RandomForestClassifier(n_estimators=400, max_features='sqrt', max_depth=40,
                                     bootstrap=False, oob_score=False, random_state=10)),
        ('SVC', SVC(C=12, probability=True)),
        ('Et', ExtraTreesClassifier(n_estimators=400, max_features='sqrt', max_depth=50)),
        ('XG', XGBClassifier(learning_rate=0.13, max_depth=5, n_estimators=300, nthread=10,
                             use_label_encoder=False, eval_metric='mlogloss'))
    ]
    # meta_classifier = LogisticRegression()
    meta_classifier = SVC(C=1, probability=True) # 使用SVC作为元分类器
    stacking_classifier = StackingClassifier(estimators=base_classifiers, final_estimator=meta_classifier)
    stacking_classifier.fit(data, label) # 训练堆叠分类器
    return stacking_classifier # 进行预测
```

```

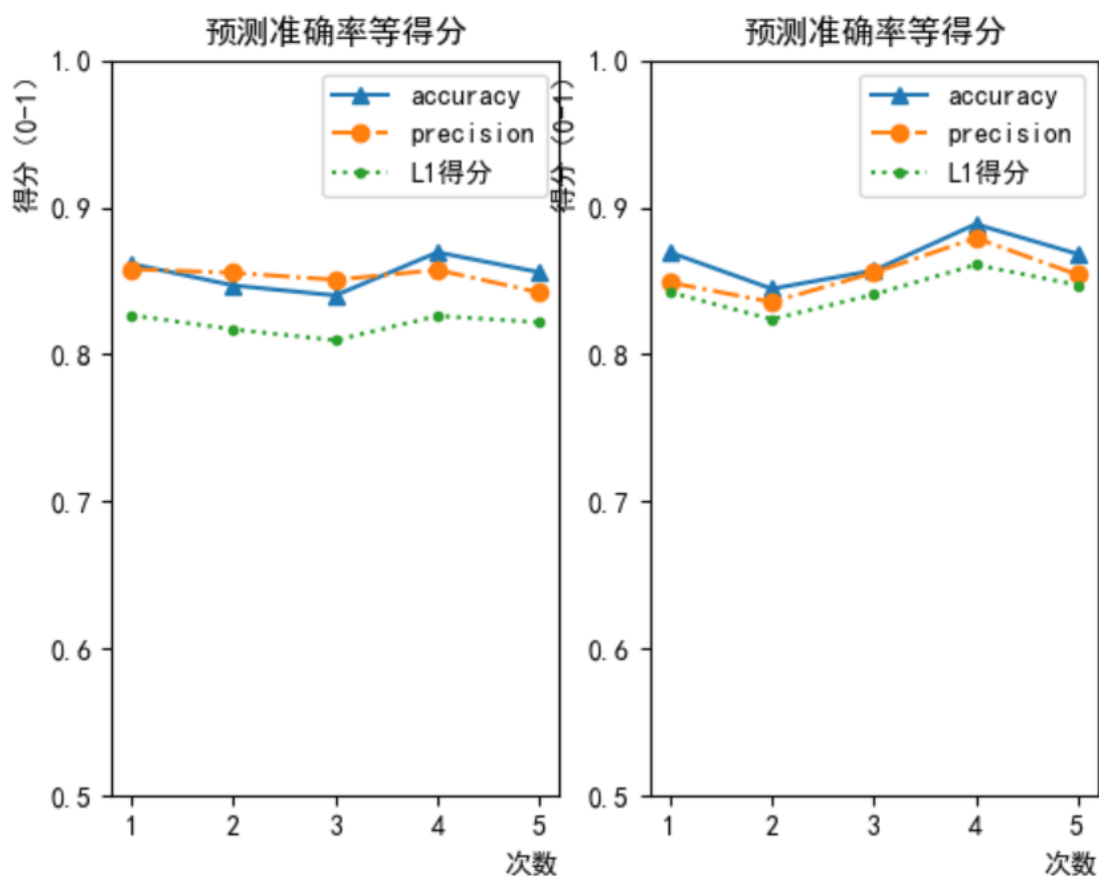
=====采用分类器学习进行分类器集成=====
正确个数: 773,总数: 888, 正确率0.870495
m_precision: 0.8603211791986244
m_recall: 0.8413403301645564
f1-score: 0.8448217651023319
=====采用分类器学习进行分类器集成=====
正确个数: 753,总数: 888, 正确率0.847973
m_precision: 0.83578644626148
m_recall: 0.8316080249256506
f1-score: 0.8296598720824561
=====采用分类器学习进行分类器集成=====
正确个数: 776,总数: 888, 正确率0.873874
m_precision: 0.8797562779915026
m_recall: 0.8525628714782525
f1-score: 0.8561796309673254
=====采用分类器学习进行分类器集成=====
正确个数: 783,总数: 888, 正确率0.881757
m_precision: 0.8633784323372087
m_recall: 0.8494611660253071
f1-score: 0.849397525126891
=====采用分类器学习进行分类器集成=====
正确个数: 776,总数: 888, 正确率0.873874
m_precision: 0.8580806950880575
m_recall: 0.8583277172717945
f1-score: 0.8567264378648046

```

## 提交时的训练情况：

	c-submission_features0.835531427734160...	2023-11-07 21:57	XLS 工作表	34 KB
	c-submission_features0.848879294572555...	2023-11-07 21:58	XLS 工作表	34 KB
	c-submission_features0.854156629948400...	2023-11-07 22:04	XLS 工作表	34 KB
	c-submission_features0.855529987962089...	2023-11-07 21:54	XLS 工作表	34 KB
	c-submission_features0.879085926082315...	2023-11-07 21:59	XLS 工作表	34 KB
	submission_features0.8421420195466678.c...	2023-11-07 21:39	XLS 工作表	34 KB
	submission_features0.8506871531882584.c...	2023-11-07 21:37	XLS 工作表	34 KB
	submission_features0.8554591360617679.c...	2023-11-07 21:36	XLS 工作表	34 KB
	submission_features0.8574004446336781.c...	2023-11-07 21:38	XLS 工作表	34 KB
	submission_features0.8578795128587364.c...	2023-11-07 21:35	XLS 工作表	34 KB

```
正确个数: 760, 总数: 888, 正确率0.855856
m_precision: 0.8421420195466678
m_recall: 0.8170609296918264
f1-score: 0.8218538108278964
=====采用分类器学习进行分类器集成=====
正确个数: 772, 总数: 888, 正确率0.869369
m_precision: 0.8488792945725555
m_recall: 0.8420472059515096
f1-score: 0.8422520199444409
=====采用分类器学习进行分类器集成=====
正确个数: 750, 总数: 888, 正确率0.844595
m_precision: 0.8355314277341602
m_recall: 0.824844393624093
f1-score: 0.8238981683252637
=====采用分类器学习进行分类器集成=====
正确个数: 761, 总数: 888, 正确率0.856982
m_precision: 0.8555299879620892
m_recall: 0.8358041169261367
f1-score: 0.840708273223418
=====采用分类器学习进行分类器集成=====
正确个数: 789, 总数: 888, 正确率0.888514
m_precision: 0.8790859260823151
m_recall: 0.8589072386138287
f1-score: 0.8609394048427722
=====采用分类器学习进行分类器集成=====
正确个数: 771, 总数: 888, 正确率0.868243
m_precision: 0.8541566299484008
m_recall: 0.8490642079362795
f1-score: 0.847118234935755
```



改进方向：实验发现 Black-grass 与 Loose Silky-bent 两类植物分类效果很差，因此选择使用对着两类进行单独实验，首先进行单独训练两类的分类情况，判断模型能否找到有效的特征进行分类。并以该两类的分类表现作为指标，进行特征提取的修改，以更好地对整个任务进行分类。

词袋大小 200

HOG特征降维 200

LBP特征降维 200

```

train文件夹文件读取中.....
100%|██████████| 1722/1722 [00:09<00:00, 191.19it/s]
100%|██████████| 1722/1722 [00:08<00:00, 197.91it/s]
0%|          | 0/1104 [00:00<?, ?it/s]test文件夹文件读取中.....
100%|██████████| 1104/1104 [00:04<00:00, 228.54it/s]
数据加载完成. train图片数量为: 1722
train_dataset: (1722, 256, 256, 3),train_label: (1722,)
test_dataset: (1104, 256, 256, 3)
0%|          | 0/2826 [00:00<?, ?it/s]1、开始进行所有图片的sift特征运算
100%|██████████| 2826/2826 [00:23<00:00, 121.08it/s]
*发现8张图片未计算出sift特征*
2.1、开始堆叠所有sift特征
shape变化: (2826, x, 128) -> (411492, 128)
2.2、开始K聚类: 聚类中心个数200, 数据点个数411492 (这个过程耗时最长)
0%|          | 0/2826 [00:00<?, ?it/s]
下面开始HOG特征提取
100%|██████████| 2826/2826 [00:53<00:00, 52.79it/s]

PCA降维中.....
48600降至200维
0%|          | 0/2826 [00:00<?, ?it/s]
下面开始LBP特征提取
100%|██████████| 2826/2826 [00:26<00:00, 108.37it/s]

PCA降维中.....
256降至200维

-----
=====采用分类器学习进行分类器集成=====
正确个数: 289,总数: 345, 正确率0.837681
m_precision: 0.8244542196155099
m_recall: 0.7499372489959839
f1-score: 0.7741301907968575
=====采用分类器学习进行分类器集成=====
正确个数: 285,总数: 345, 正确率0.826087
m_precision: 0.8435540248792022
m_recall: 0.7549397314617028
f1-score: 0.7776679841897234
=====采用分类器学习进行分类器集成=====
正确个数: 296,总数: 344, 正确率0.860465
m_precision: 0.8726771436448856
m_recall: 0.7786435786435786
f1-score: 0.8078570098678086
=====采用分类器学习进行分类器集成=====
正确个数: 290,总数: 344, 正确率0.843023
m_precision: 0.8376433785192909
m_recall: 0.7654098360655737
f1-score: 0.7890529184646831
=====采用分类器学习进行分类器集成=====
正确个数: 296,总数: 344, 正确率0.860465
m_precision: 0.8300653594771241
m_recall: 0.7869318181818181
f1-score: 0.8045454545454546

```