工作记录

1.常用分类器测试和使用

数据集：mnist

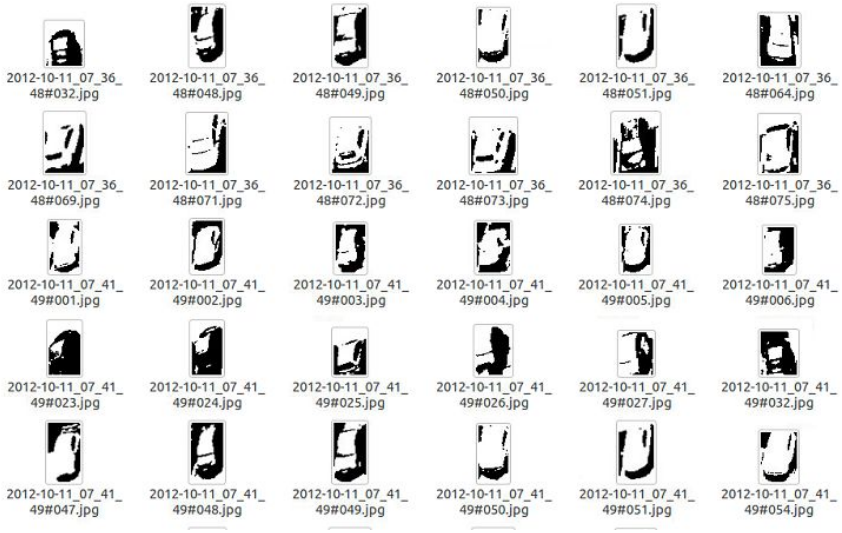
```
******************** Data Info *********************
#training data: 50000, #testing_data: 10000, dimension: 784
******************** NB ********************
training took 0.156519s!
accuracy: 83.69%
******************** KNN ********************
training took 6.642330s!
accuracy: 96.64%
******************** LR ********************
training took 50.204328s!
accuracy: 91.99%
******************** RF ********************
training took 5.153359s!
accuracy: 94.02%
******************** DT ********************
training took 38.914205s!
accuracy: 87.06%
******************** SVM ********************
training took 2205.018649s!
accuracy: 94.35%
******************** GBDT ********************
training took 16436.225556s!
accuracy: 96.18%
gnss@gnss:~/devdata/Data/PKLot/testcar$
```

代码:

```
if __name__ == '__main__':
    data_file = "mnist.pkl.gz"
    thresh = 0.5
    model_save_file = None
    model_save = {}

    test_classifiers = ['NB', 'KNN', 'LR', 'RF', 'DT', 'SVM', 'GBDT']
    classifiers = {'NB':naive_bayes_classifier,
                   'KNN':knn_classifier,
                    'LR':logistic_regression_classifier,
                    'RF':random_forest_classifier,
                    'DT':decision_tree_classifier,
                   'SVM':svm_classifier,
                 'SVMCV':svm_cross_validation,
                  'GBDT':gradient_boosting_classifier
    }

    print 'reading training and testing data...'
    train_x, train_y, test_x, test_y = read_data(data_file)
    num_train, num_feat = train_x.shape
    num_test, num_feat = test_x.shape
    is_binary_class = (len(np.unique(train_y)) == 2)
    print '******************** Data Info *********************'
    print '#training data: %d, #testing_data: %d, dimension: %d' % (num_train, num_test, num_feat)

    for classifier in test_classifiers:
        print '******************** %s ********************' % classifier
        start_time = time.time()
        model = classifiers[classifier](train_x, train_y)
        print 'training took %fs!' % (time.time() - start_time)
        predict = model.predict(test_x)
        if model_save_file != None:
            model_save[classifier] = model
        if is_binary_class:
            precision = metrics.precision_score(test_y, predict)
```

2. PKlot 数据处理

a） 二值化处理



b） 归一化处理

处理为 40*60 的大小

c） 存为 lmdb 形式，计算均值 mean.npy