

# Project5 手写体识别运行文档

## 目录

Project5 手写体识别源码文档 .....	1
项目文件结构.....	1
项目概述.....	2
源码说明.....	2
项目运行步骤.....	4

## 项目文件结构

```
| .gitignore
| handwriting.png
| README.md
| requirements.txt
├──handwriting_classify
|   | checkpoint
|   | CNN_model.ckpt.data-00000-of-00001
|   | CNN_model.ckpt.index
|   | CNN_model.ckpt.meta
|   | cnnPainter.py
|   | dataset.py
|   | label_data.npy
|   | number_data.npy
|   | RestoringOneCNNmodel.py
|   └──__pycache__
|       dataset.cpython-36.pyc
├──Mnist
|   mnist_bp.py
|   mnist_cnn.py
|   mnist_NaiveBayes.py
|   mnist_svm.py
├──MNIST_data
|   t10k-images-idx3-ubyte.gz
|   t10k-labels-idx1-ubyte.gz
|   train-images-idx3-ubyte.gz
|   train-labels-idx1-ubyte.gz
├──Ourdata
|   | main.py
|   | mydata_NaiveBayes.py
```

```

| | mydata_svm.py
| | mydata_Tensor_BP.py
| | mydata_Tensor_CNN.py
| |─data
| | | collect_dataset.py
| | | label_data.npy
| | | number_data.npy
| | | ourdataset.py
| | |─__pycache__
| | |   ourdataset.cpython-36.pyc
| |─model
| |   BP_model.ckpt.data-00000-of-00001
| |   BP_model.ckpt.index
| |   BP_model.ckpt.meta
| |   checkpoint
| |   CNN_model.ckpt.data-00000-of-00001
| |   CNN_model.ckpt.index
| |   CNN_model.ckpt.meta
| |   NaiveBayes_model.model
| |   svm_model.model
|─__pycache__
  dataset.cpython-36.pyc

```

## 项目概述

本次项目中我们选择的是Project5分类问题，Project5中给出的数据来源是<http://ai.berkeley.edu/classification.html>，但是原伯克利大学的项目中给出的数据集只有100条左右，算是toy-dataset。

所以征得老师同意以后，我们选择了手写数字识别的经典数据集：**MNIST**手写体数据集。并且我们还确定了：本次项目中我们选择手写体识别这一个任务并至少实现四种不同的算法完成任务。于是，本次实验中我们分别实现：朴素贝叶斯、支持向量机、BP神经网络、CNN卷积神经网络这几种算法，将MNIST数据集分割成train、test两个数据集，先在train数据集上训练模型然后在test数据集上测试识别精度。

除使用MNIST数据集测试算法以外，我们另外自行实现了一套从收集手写体数据集，到训练，再到在电脑上实时演示手写数字识别的工具。这一套工具的全部源码同样包含在了项目中。

## 源码说明

### 1. /requirements.txt

储存了本项目依赖的第三方 python 包。

### 2. /MNIST\_data/

本文件夹中储存了MNIST数据集，分别是：

train-images-idx3-ubyte.gz

```
train-labels-idx1-ubyte.gz
t10k-images-idx3-ubyte.gz
t10k-labels-idx1-ubyte.gz
```

其中前两个文件分别是MNIST训练集的手写体图片、对应数字标签数据集，后两个文件是MNIST测试集的手写体图片、对应数字标签数据集。前者大约有55000张手写体数字图片以及对应数字，图片分辨率 $28 \times 28$ ，后者大约有10000张手写体图片及对应数字。

### 3. /Mnist/

本文件夹存我们使用4种算法训练并识别MNIST数据集的代码。

#### 3.1 /Mnist/mnist\_bp.py

BP神经网络训练、识别MNIST手写体数据集代码。

#### 3.2 /Mnist/mnist\_cnn.py

CNN卷积神经网络训练、识别MNIST手写体数据集代码。

#### 3.3 /Mnist/mnist\_NaiveBayes.py

朴素贝叶斯算法训练、识别MNIST手写体数据集代码。

#### 3.4 /Mnist/mnist\_svm.py

SVM支持向量机训练、识别MNIST手写体数据集代码。

### 4. /Ourdata/

本文件夹存我们实现的手写体识别程序，能够实现：

- a) 在电脑上自行收集个人的手写体数字数据集
- b) 使用BP神经网络、CNN神经网络、朴素贝叶斯算法、SVM算法，在自己收集的数据集上训练识别模型，并将训练好的模型保存下来。
- c) 使用自行训练的模型，在电脑上手写数字，然后实时输出不同算法识别结果

#### 4.1 /Ourdata/data/

本文件夹储存手写体数据集的收集以及读取代码，也存有收集的数据集。

##### 4.1.1 /Ourdata/data/collect\_dataset.py

收集手写体数据集。

##### 4.1.2 /Ourdata/data/ourdataset.py

读取收集好的数据集。

##### 4.1.3 /Ourdata/data/number\_data.npy

收集到的手写体图片数据，图片大小 $28 \times 28$ ，使用numpy包的二进制格式储存。目前已经收集到了417个自己写的数字样本。

##### 4.1.4 /Ourdata/data/label\_data.npy

手写体图片对应数字标签。同上，已收集到417个样本。

#### 4.2 /Ourdata/data/model/

本文件夹储存训练好的，各个算法的模型。

#### 4.3 /Ourdata/mydata\_NaiveBayes.py

使用自行收集的手写体数据训练朴素贝叶斯模型，并保存。

#### 4.4 /Ourdata/mydata\_svm.py

使用自行收集的手写体数据训练SVM模型，并保存。

#### 4.5 /Ourdata/mydata\_Tensor\_BP.py

使用自行收集的手写体数据训练BP神经网络，并保存。

#### 4.6 /Ourdata/mydata\_Tensor\_CNN.py

使用自行收集的手写体数据训练CNN卷积神经网络，并保存。

#### 4.7 /Ourdata/main.py

在电脑上演示实时识别手写数字的主程序。

## 项目运行步骤

### 1. 测试 MNIST 数据集

命令行窗口目录应在本项目的**第一层文件夹下**，使用以下命令测试各个算法识别 MNIST 手写数字数据集的效果

#### 1.1 测试 NaiveBayes

```
python ./Mnist/mnist_NaiveBayes.py
```

#### 1.2 测试 SVM

```
python ./Mnist/mnist_svm.py
```

#### 1.3 测试 BP 神经网络

```
python ./Mnist/mnist_bp.py
```

#### 1.4 测试 CNN 神经网络

```
python ./Mnist/mnist_cnn.py
```

### 2. 测试实时手写体识别

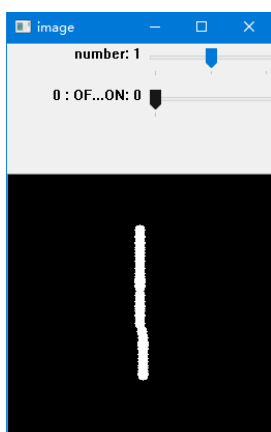
#### 2.1 收集手写数据

a) 运行命令 `python ./data/collect_dataset.py`

b) 出现如下图所示窗口：



c) 该窗口中 `number` 滑块可以在 0-9 间变化，代表输入样本正确数字，黑色区域可以写字。按住鼠标左键在黑色区域可以写数字，松开左键不再写字。最后按一下鼠标右键确认当前样本数字，样本图片及对应数字被记录下来，一次样本采集结束，黑色画板清空。如图：



- d) 重复上一步骤，并改变 **number** 滑块可以输入不同的数字样本。
- e) 样本收集结束后不要用“X”按钮关闭窗口，按 ESC 键退出，同时才会保存输入的所有样本。
- f) 下一次输入命令再次运行时，自动在上一次收集数据基础上添加样本。如果想重新收集数据集，可以删除 **number\_data.npy** 和 **label\_data.npy** 文件，再从第一个步骤开始。

## 2.2 训练模型

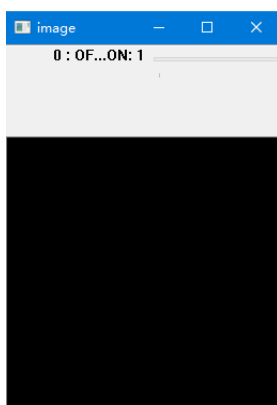
在收集足够多的不同数字的手写体数据以后可以训练算法模型。

- a) 使用命令 `python ./Ourdata/mydata_NaiveBayes.py` 训练朴素贝叶斯模型，模型会自动保存下来。
- b) 使用命令 `python ./Ourdata/mydata_svm.py` 训练 SVM 模型
- c) 使用命令 `python ./Ourdata/mydata_Tensor_BP.py` 训练 BP 神经网络模型
- d) 使用命令 `python ./Ourdata/mydata_Tensor_CNN.py` 训练 CNN 卷积神经网络模型

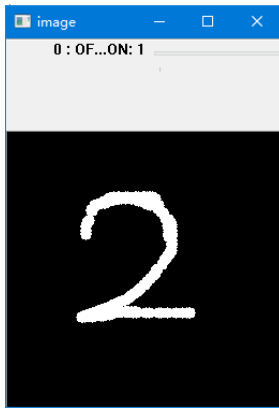
## 2.3 运行手写数字实时识别程序

训练完所有的算法模型后，可以开始使用演示程序。

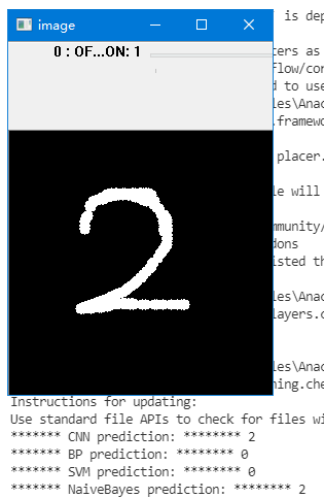
- a) 运行命令 `python ./Ourdata/main.py`
- b) 出现如下窗口：



- c) 此时只有 **OF...ON** 滑块开关显示为 1 时才可以在黑色画板上写字，写字方法同收集时的方法。如图所示：



- d) 写完一个数字后，在黑色画板点一下鼠标右键则识别当前所写数字，各个算法识别结果输出在终端中，如图：



- e) 识别一次后画板不会清空，方便修改。可以把 OF...ON 滑块置为 0，再置为 1，则画板清空，且可以再次写数字。
- f) 最后按 ESC 键退出演示。