# 基于最小错误率的贝叶斯决策实现手写数字识别

## 实验目的

建立基于最小错误率的贝叶斯决策分类器，实现手写数字的识别。

## 实验方法

### 搭建平台

使用Windows系统自带的画图工具分别构建训练样本集和测试样本集，使用MATLAB GUI编程环境对它们进行处理、识别。

### 特征描述

将手写部分从图像中分割，并转化为灰度图像，而后将其分割为6\*6=36个小图片，用每个小图片中黑色像素占总像素的比重作为该样本的特征，也就是一个36维的向量。

子函数get\_feature(image)代码如下：

function[feature] = get\_feature(image)

%从图像中分割出手写部分并转为灰度

image2 = pretreat(image);

A = mat2cell(image2,[10,10,10,10,10,10],[7,7,7,7,7,7]);

B = reshape(A,1,36);

for i=1:length(B)

temp = cell2mat(B(i));

%feature(i)表示黑像素点占一个小块的比例

feature(i) = sum(temp(:) ~= 255) / numel(temp);

end

%feature

end

### 关键算法

本次实验采用最小错误率贝叶斯决策，最小错误率就是求解一种决策规则，使错误率最小。

即，而,也就是说，使错误率最小的决策就是使正确率最大的决策。

由贝叶斯公式知，要求后验概率关键在求类条件概率密度函数。

本次实验中关键代码如下：

%最小错误率求类条件概率密度函数pXwi

function[pXwi] = get\_pXwi(new\_feature)

mean\_feature = [];

for i = 0:9

mean\_feature = [mean\_feature;get\_mean\_feature(i)];

end

%mean\_feature有10行，每一行代表一个类的特征

%二值化

new\_feature = new\_feature>0.4;

pXwi = [];

for i=0:9

p = 1;

for j =1:length(new\_feature)

if(new\_feature(j))

p = p\*mean\_feature(i+1,j);

else

p = p\*(1-mean\_feature(i+1,j));

end

end

pXwi = [pXwi,p];

end

## 实验结果

### 训练样本集构建

使用Windows系统自带的画图软件创建训练样本集，使用10个文件夹分别保存10个类的训练集，如下图。



图 三-1 训练样本集

每一个样本都有一个固定的模板图片，用画图工具在该图片的指定位置手写数字，模板图片如下图。

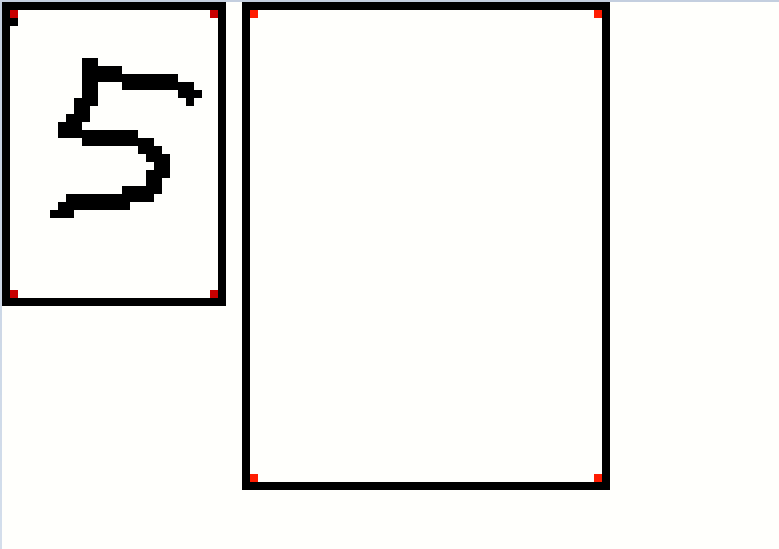


图 三-2 样本构建的模板图片

某一手写完成的样本如下图。

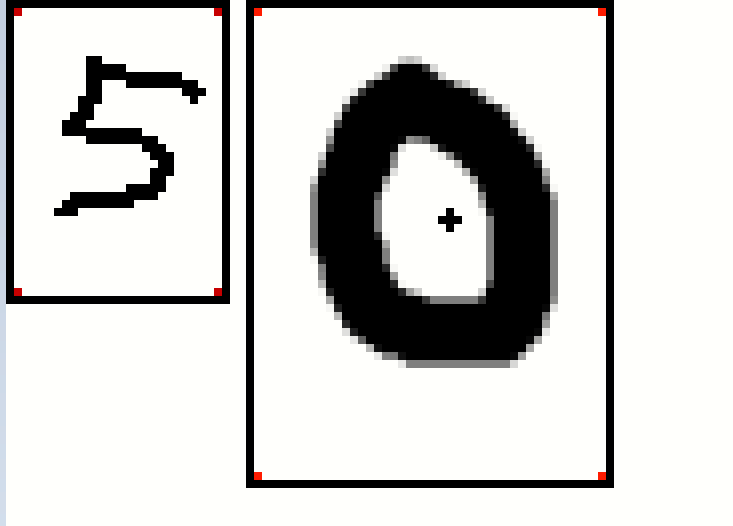


图 三-3 某一手写完成的样本

### 测试集构建

测试集样本的制作与训练集基本相同，不同的是，将所有图片都保存至同一文件夹“测试样本集”中，以供程序读取识别。

本次实验测试样本集如下图。

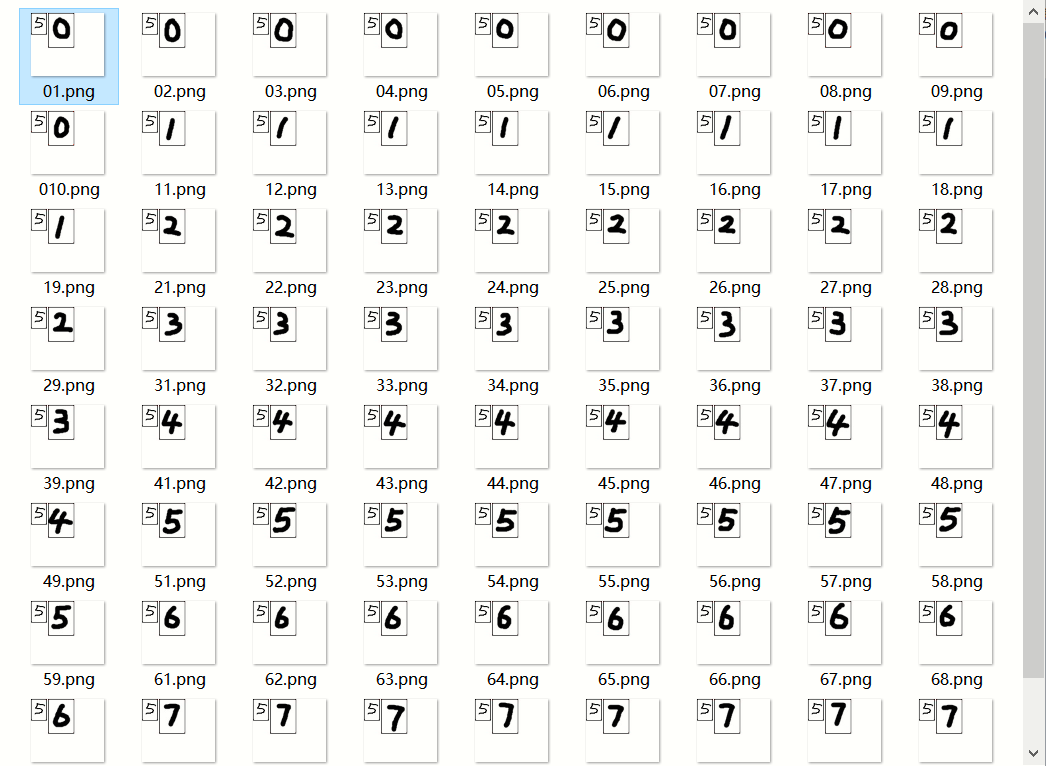


图 三-4 测试样本集

该样本集中有手写数字0-9各10个，它们的命名以各自的数字开头，用于实验中计算正确率。

### 实验结果

实验的运行环境是MATLAB GUI，本次实验各控件布局如下图。

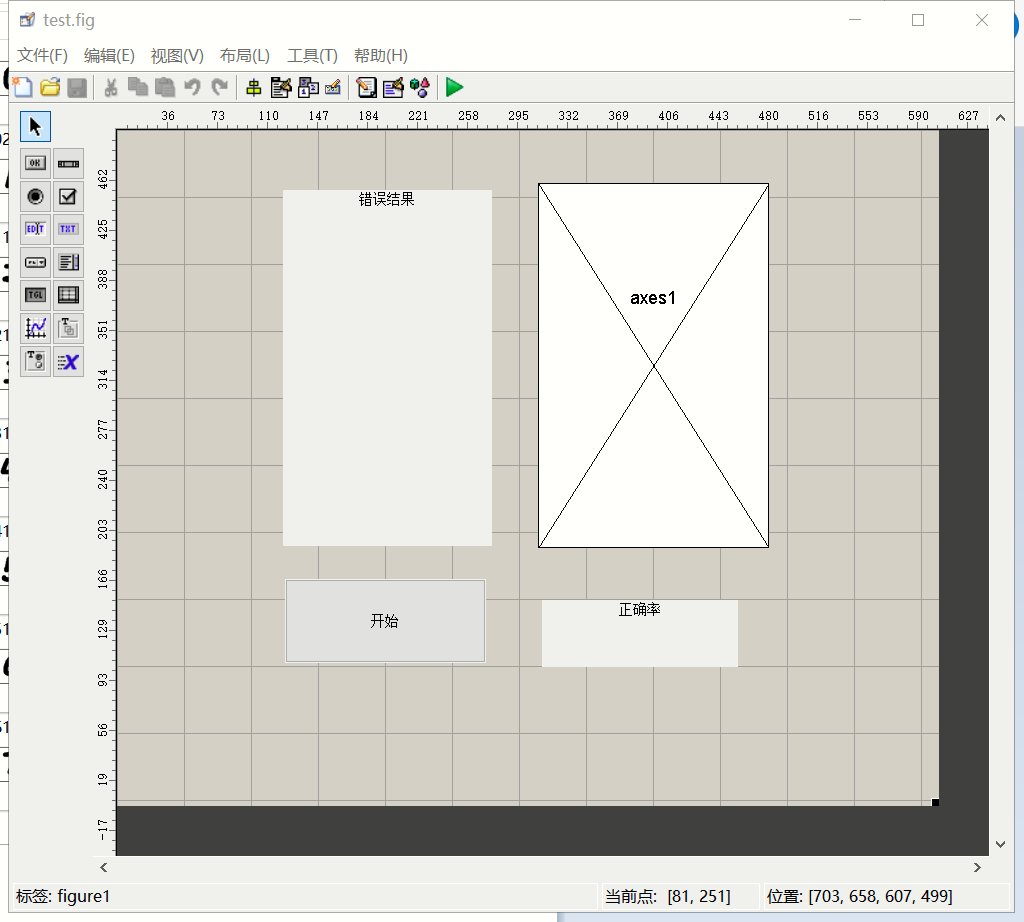


图 三-5 GUI控件的布局

图中已表明了各控件的作用，其中axes1用于运行时显示正在识别的样本。

下图为程序正在运行。

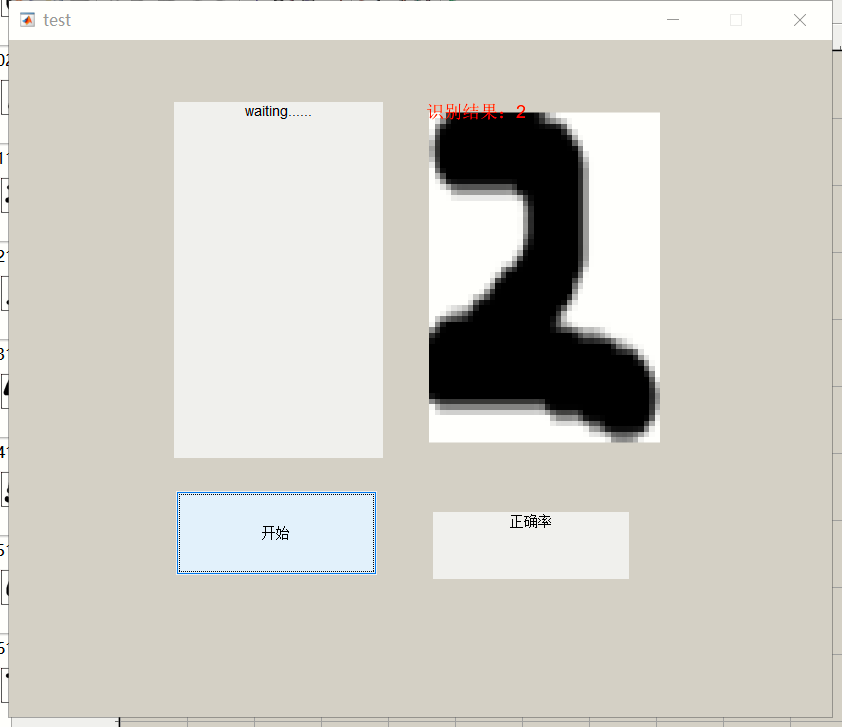


图 三-6 程序正在运行

程序运行结束时如下图。

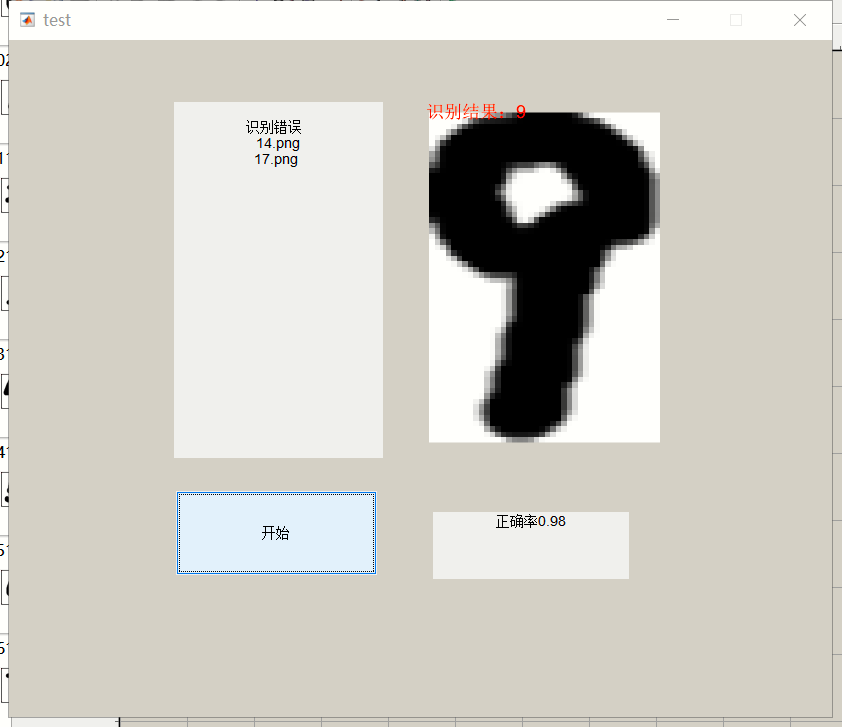


图 三-7 程序运行结束

运行结束时显示出了识别错误的测试样本以及正确率。

### 性能量化评估

运行结果显示，本次实验中识别的正确率为98%。

运行过程中发现，对每个样本识别速度较慢，原因是程序最初是针对单个样本识别设计的，所以在每次识别的时候都对训练样本进行了读取，做了很多不必要的工作，导致运行时间长，下一次试验将重新设计以改进这个不足。

## 结果讨论

结果显示有两个样本识别错误，分别是“14.png”和“17.png”，它们都是数字1的待识别样本。

在命令行中对识别错误的样本进行分析：

>> image = imread('./测试样本集/14.png');

>> new\_feature = get\_feature(image);

>> pXwi = get\_pXwi(new\_feature);

>> pwiX = get\_pwiX(pXwi);

得到各类的后验概率为pwiX=[0,0.0184,0,0,0,0.9816,0,0,0,0]，表示程序认为该样本是5的概率极大（对另一错误样本测试也得到了类似结果）。

通过对其他同类样本（数字1）的观察，发现错误样本确实“与众不同”，直观上看，其他同类样本（数字1）都有一定的“倾斜度”，而错误样本不然。通过后续增加训练样本数量应该可以提高正确率。

## 结论

本次实验使用MATLAB编程环境完成了基于最小错误率的贝叶斯决策算法的编写，实现了手写数字的识别，通过对测试样本集（100个样本）的识别，结果显示正确率达98%，表示实验过程中各个样本集的建立以及所用算法较为可靠。