## 综合图像处理系统

###### 一、用户界面如下所示：

此界面通过MATLAB编写，通过三个面板来显示图像，首先点击“选择图像”，在原始图像框中显示；然后选择要处理的方式，点击前方按钮，在过渡图像相框和处理结果框中显示过渡图像和最终图像。

主要功能如下：

1. 图像基本处理，包括图像去噪、直方图均衡化。
2. 图像边缘检测，包括使用canny算子进行边缘检测，Hough变换进行直线检测。
3. 图像分割处理，包括使用最大类间方差法、K-means均值法、Otsu方法分别对图像进行分割处理。
4. 图像目标检测与识别，包括车牌识别和物体检测。
5. 特征提取，包括SIFT特征提取，SURF特征提取与匹配。
6. 主成分分析法进行人脸识别，包括两个文件夹：test、train，以及整个test文件夹下所有图识别准确率。
7. 卷积神经网络集实现MNIST手写数据集训练。

###### 二、实现过程和结果

1. 图像基本处理

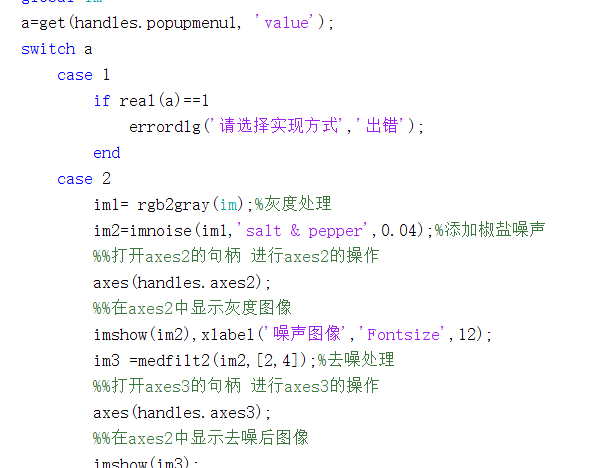
（1）图像去燥

运行演示：



实现代码：

首先将原始图像进行灰度处理，然后添加椒盐噪声，最后通过中值滤波对图像进行去噪处理。



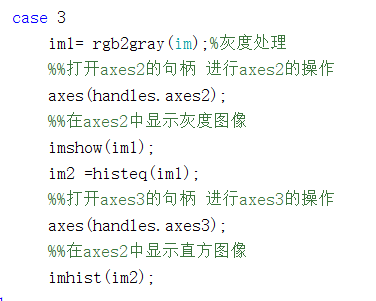
（2）直方图均衡化

运行演示：



实现代码：

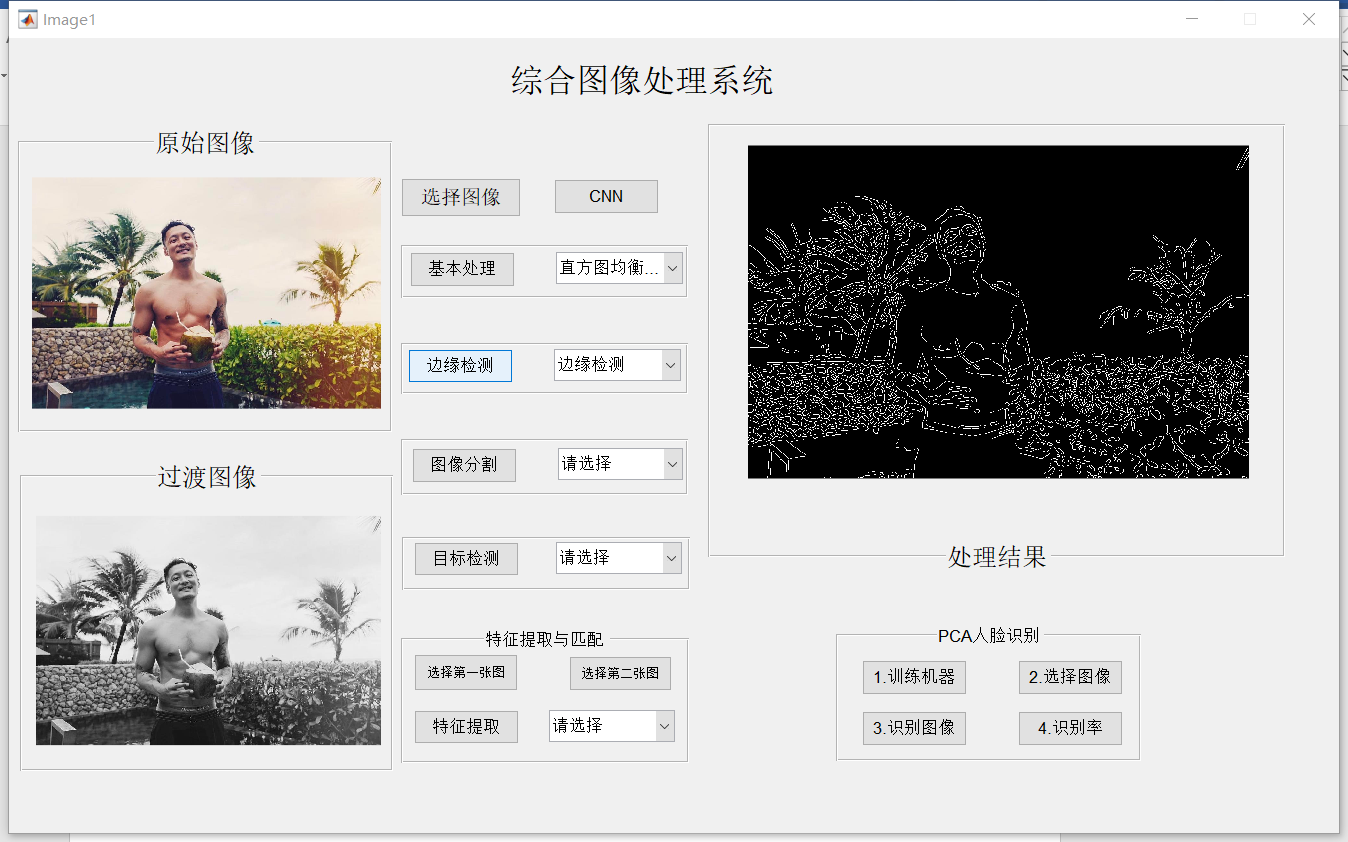
首先对原始图像进行灰度处理，然后通过hist()函数把原始图像的灰度统计直方图变换成为均匀分布的形式，然后通过imhist()函数提取图像中的直方图信息并显示在处理结果框中。



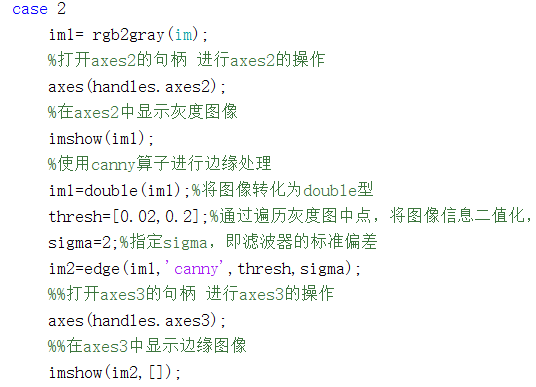
1. 图像边缘检测

（1）边缘检测

运行演示：



实现代码：首先将原始图像进行灰度处理，然后将灰度图像转化为double型，最后使用edge（）函数用canny算子进行边缘检测。



（2）Hough变换

运行演示：



实现代码：首先同边缘检测一样，使用canny算子对图像进行边缘描绘，将边缘图像显示在过渡图相框中；然后使用hough()进行霍夫变换，使用houghpeaks()进行霍夫变换峰值检测；最后使用houghlines（）函数进行霍夫变换线检测，并将将线段绘制在图像上，并在处理结果框中显示。

实现代码：



1. 图像分割

（1）最大类间方差法

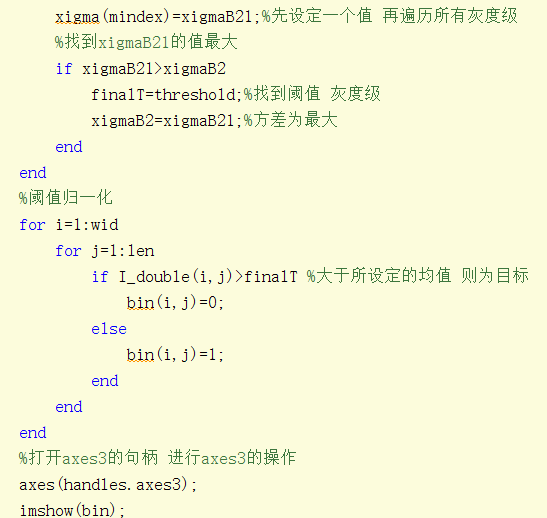
运行演示：



代码实现：

首先将原始图像转化为灰度图，计算图像的直方图并将直方图归一化；然后设置阈值，计算目标和背景的灰度平均值，找到最大方差并进行阈值归一化，将大于阈值的设为设为目标。



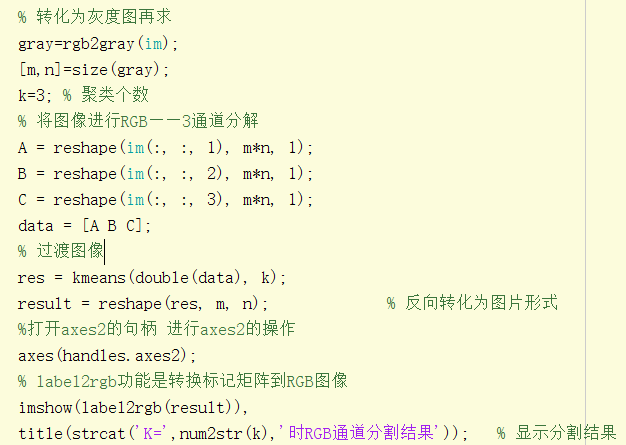


（2）K均值聚类算法

运行演示：



实现代码：首先对原始图像进行灰度转化，求图像大小并设置聚类个数；然后将图像进行RGB三通道分解，使用kmeans()函数将分解后的图像矩阵划分为K个类；最后使用reshpe()函数反向转化为图像格式，再使用label2rgb()函数转换为RGB图像。



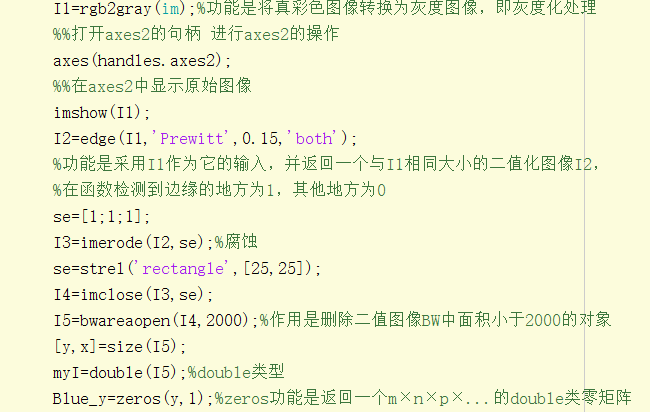
1. 目标检测

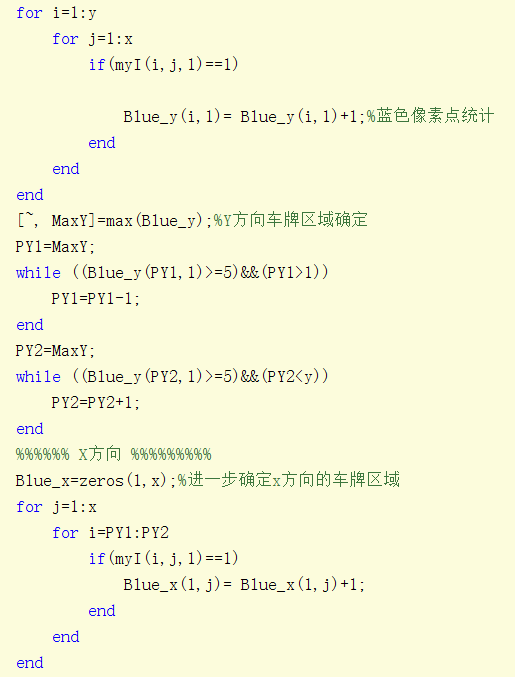
（1）车牌识别

运行演示：



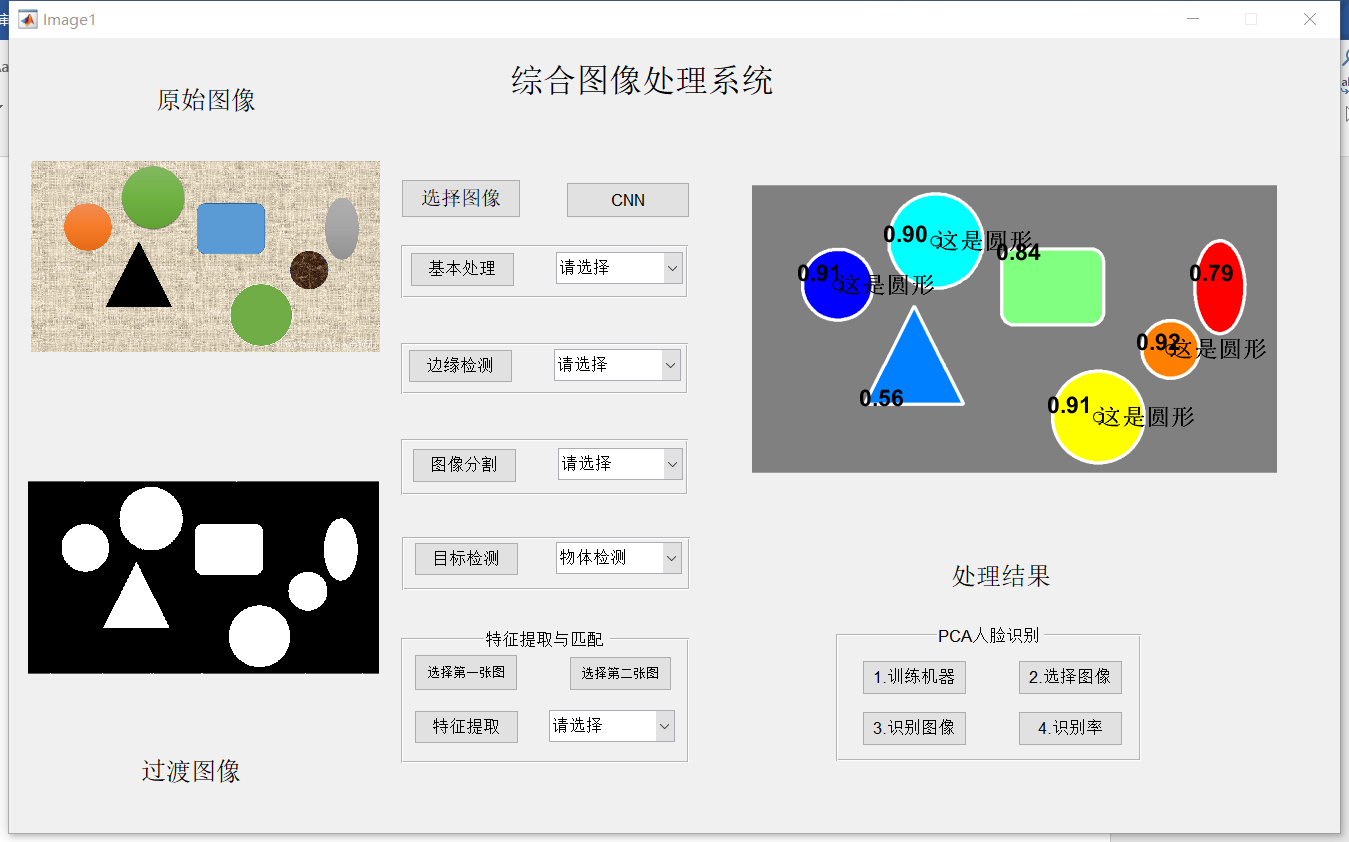
实现代码：



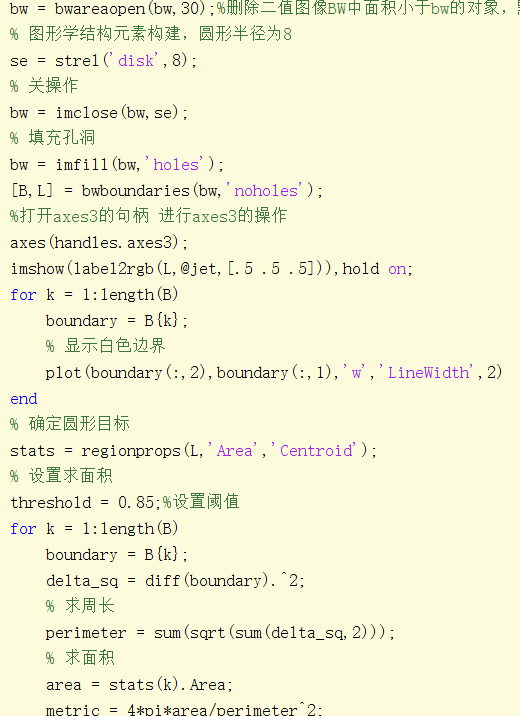
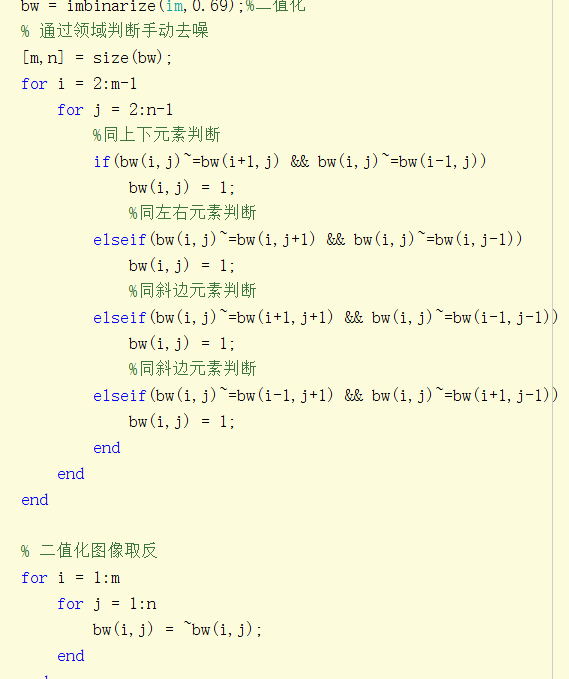


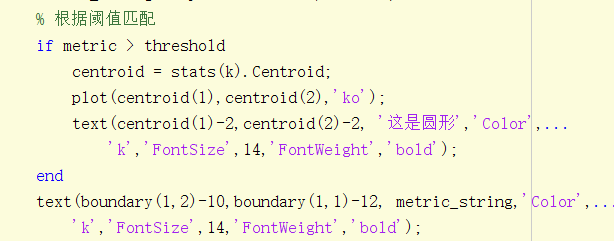
（2）物体检测

运行演示：



实现代码：首先对原始图像进行灰度处理，转化为灰度图像，再将灰度图像转化为二值图像；然后对二值图像进行去噪和取反，找出圆形目标；最后计算其周长和面积，求得圆形。





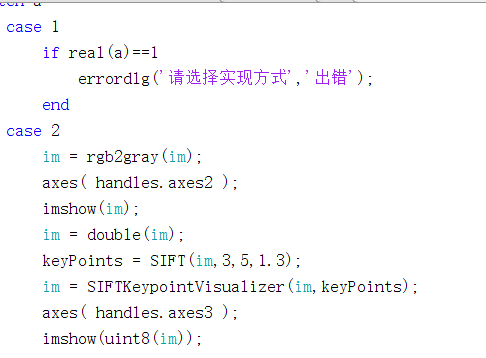
1. 特征提取与匹配

（1）SIFT特征提取

运行演示：首先点击“选择图像”选择一张图像，然后选择SIFT特征提取，点击“特征提取”按钮。



实现代码：首先将原始图像转换为灰色图像并显示在过渡图像框中，调用SIFT.m文件提取sift特征，然后调用SIFTKeypointVisualizer.m文件创建图像的可视化sift特征，keypoint.m文件用来用SIFT提取关键点。



（2）SURF特征提取与匹配：

首先点击“选择第一张图”选择一张图像，然后点击“选择第二张图”选择第二张图，效果如图表1所示；点击选择“SURF特征提取与匹配”，最后点击“特征提取”，效果如图表二所示。



图表 1



图表 2

实现代码：



1. PCA人脸识别

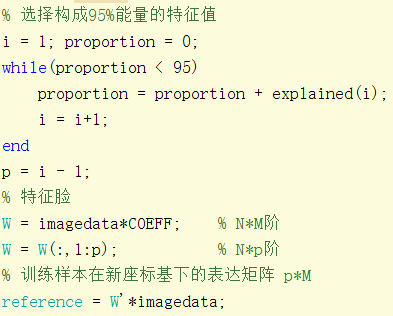
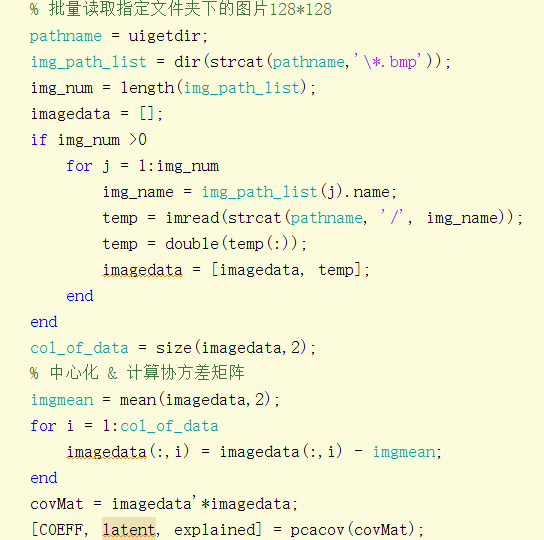
运行演示：

首先点击“训练机器”选择train文件夹，其次点击“选择图像”在test文件夹中选择一幅图像，然后点击“识别图像”，会在train文件夹中选择一幅匹配度最高的图像显示；最后点击“识别率”选择test文件夹，会显示整个文件中的图像与train文件夹中的图像匹配的识别率。

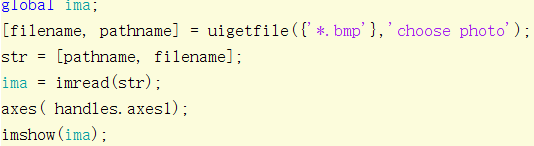


实现代码

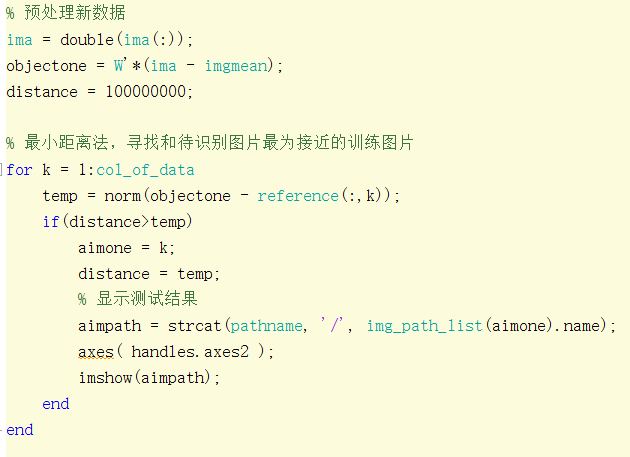
1. “训练机器”代码：



1. “选择图像”代码：



1. “识别图像”代码:



1. “识别率”代码：

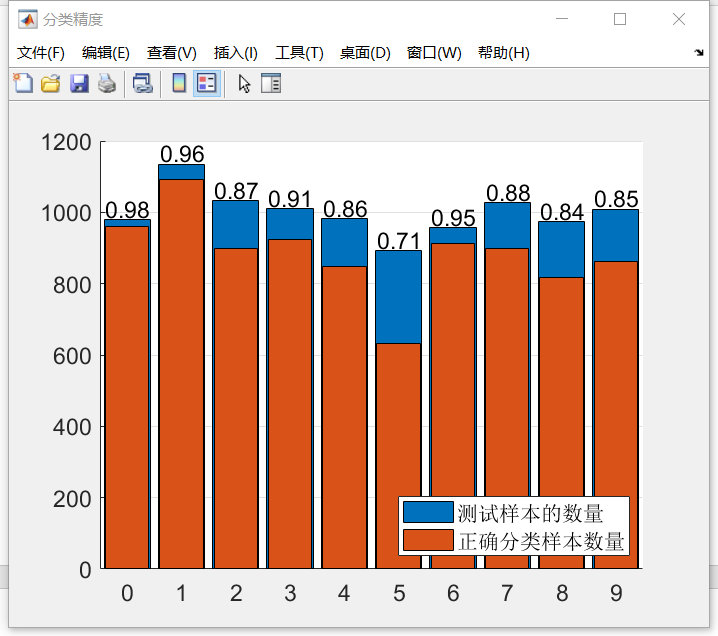
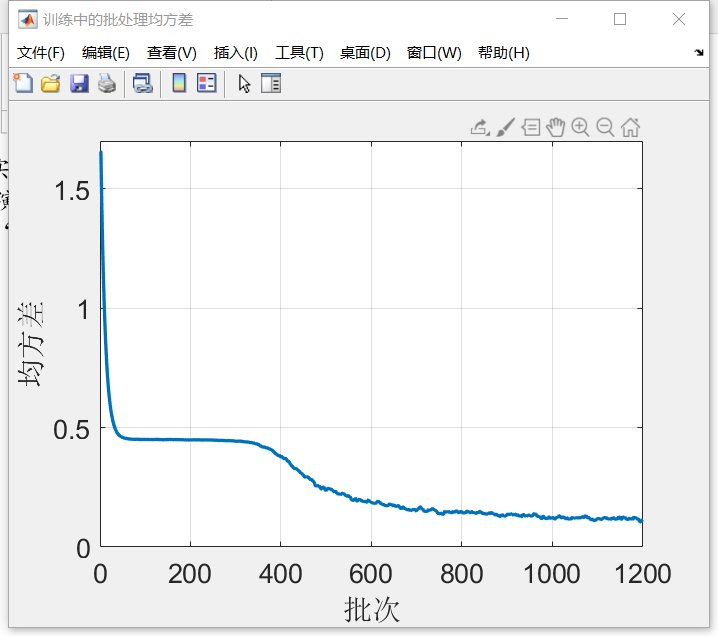


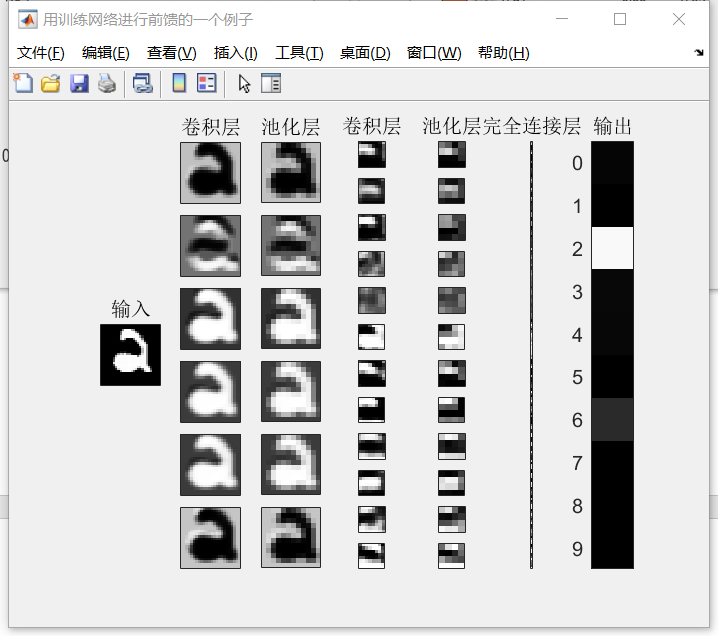
1. CNN实现MNIST手写数据集识别

运行演示：

点击“CNN”会显示：







实现代码:

首先加载mnist数据集并格式化数据集，然后建立卷积神经网络，调用cnntrain.m文件训练cnn,调用cnntest.m测试cnn。最后画出每批训练过程中的误差均方图、分类精度、卷积核的可视化、以及用训练网络进行前馈的一个实例。

