噪声手写数字图片中手写数字识别系统 V1.0

使用说明书

1.系统简介

1.1 系统搭建背景

现在是读图时代，我们每天遇到大量的含有手写数字的图片。手写数字识别的意义在于将传统的印刷体数字和手写数字转化为机器可处理的形式，实现自动化数据处理、手写信息检索、提高效率和准确性等方面的应用。它对于各种行业和领域来说，都具有重要的实用价值和发展潜力。如何从图像中自动获取并理解数字信息就是一个实际问题。并且大部分需要识别的图片都伴随有噪声干扰，需要先将噪声干扰尽可能减少再进行识别。

本系统可以实现噪声影响下的手写数字识别分类，使用者先在系统内完成识别模型的训练，即可选择一个噪声手写识别图片进行识别，并在系统消息框得到识别结果。

1.2 系统开发环境介绍

系统开发及测试环境见表 1.1：

表 1.1 开发及测试环境

硬件配置	
CPU	Intel Core i9-13900HX
内存	32GB
软件平台	
操作系统	Windows 11 22H2
开发环境	Matlab 2023a

1.3 系统功能设计

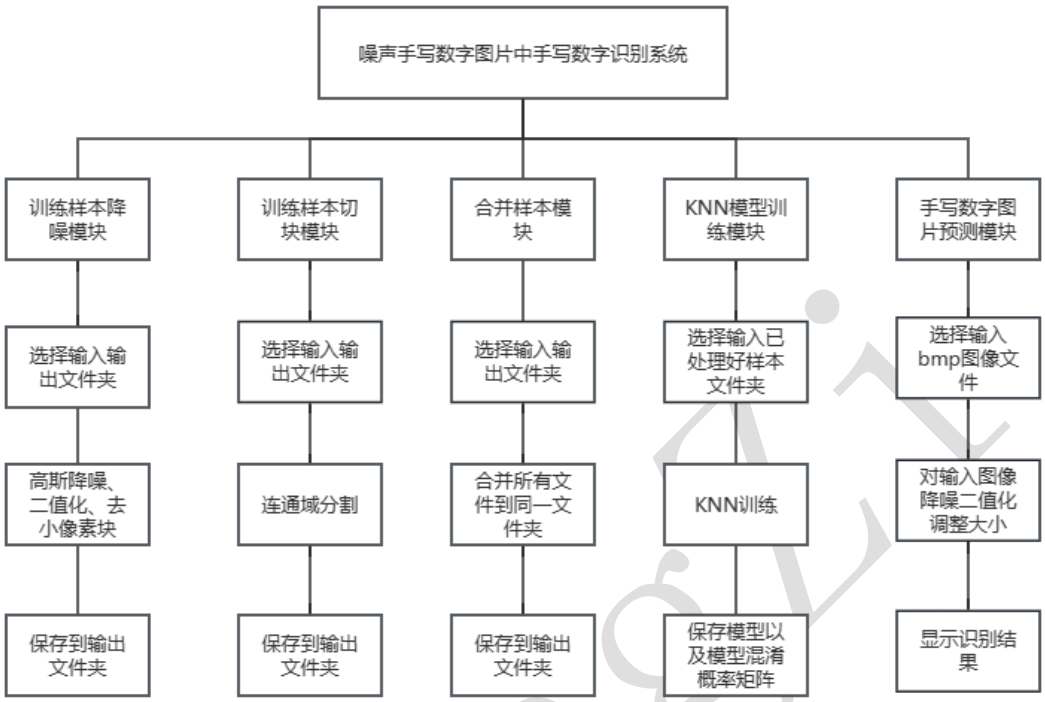

图 1.1 系统功能模块

1）训练样本降噪模块

此模块在用户点击“训练样本降噪处理”按钮后，选择输入输出的训练样本文件夹，程序开始对所有样本图片进行高斯降噪、二值化、去小像素块处理，将处理好样本图片保存到输出文件夹。

2）训练样本切块模块

本模块可实现将前一步得到样本切块处理。用户点击“训练样本切块处理”后选择输入输出文件夹就会将切割好的样本图片保存在样本同名文件夹中并按编号排好。

3）合并样本模块

此模块将前一模块得到的不同文件夹的样本合并到同一文件夹。用户点击“合并样本处理”按钮并选取输入输出文件夹路径后，将执行合并操作。

4）KNN 模型训练模块

此模块将前一模块得到处理好的训练样本进行 KNN 训练得到可用于识别的模型。用户点击“KNN 模型训练”按钮并选取输入文件夹路径后，将自动划分训练集验证集进行 KNN 训练，并将得到的模型以及模型混淆概率矩阵保存到程序同一路径下。

5）手写数字图片预测模块

此模块需要有前面模块训练好的 KNN 模型，用户点击“手写数字图片预测”后，选择需要识别的手写数字图片，模块将先进行降噪二值化以及调整大小处理，再与模型比对特征，最后在消息框显示识别结果。

1.4 噪声手写数字图片中手写数字识别算法介绍

1.4.1 算法流程

以 Matlab 为平台，通过二维滤波器设计、目标定位与分割、图像特征提取、分类器设计、数字识别这五步，实现手写数字识别分类。

二维滤波器设计（训练样本降噪）：以数字信号处理中 DFT 和 FFT 的基本原理、常见的经典滤波器设计原理为基础；在此基础上将 DFT 拓展到二维，设计经典的二维滤波器，完成图像去噪任务；对所有样本图片进行高斯降噪、二值化、去小像素块处理，将处理好样本图片保存到输出文件夹。

目标定位与分割（训练样本切块）：根据字符的特点，用一个矩形定位出其在图像中的位置，在此基础上将目标字符图像从原图像中分割出来；用检测连通域的方法，通过算法提取每个连通域（白色像素数字字符区域）的左右上下边界位置，并将其切割为单独的图像，保存到文件夹，再通过合并模块合并为下几步需要的文件夹格式。

图像特征提取（KNN 模型训练）：将分割出来的字符进行特征提取，比如对字符图像进行固定大小的划分，根据像素二值矩阵特征，得到表示该字符的一个向量；读取训练样本后自动提取其像素二值矩阵特征，依次提取所有样本图片特征

分类器设计（KNN 模型训练）：熟悉线性分类器，以 K 近邻分类器的基本原理为基础。在此基础上设计手写数字识别分类器。读取训练样本后自动划分训练集验证集进行 KNN 训练，并将得到的模型以及模型混淆概率矩阵保存到程序同一路径下。

手写数字识别：读取新手写数字图像，对其进行降噪二值化调整大小处理，提取其特征与训练好的模型比对、得到识别结果。并在程序中显示识别结果。

整体算法流程如图 1.2 所示，包括了二维滤波器设计（训练样本降噪）、目标定位与分割（训练样本切块）、图像特征提取（KNN 模型训练）、分类器设计（KNN 模型训练）、手写数字识别几个部分。最终实现在交互界面选择文件后，点击手写数字识别后可以在界面上显示识别结果。

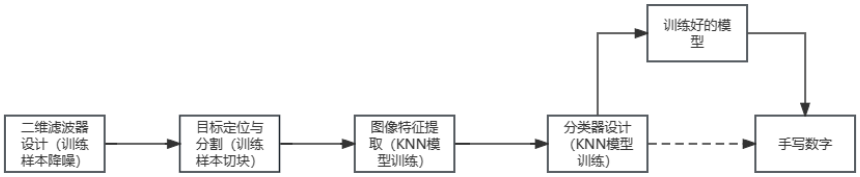

图 1.2 算法流程

1.4.2 二维滤波器设计（训练样本降噪）

我使用高斯滤波算法为原理，编写了二维低通滤波器，实现了实验的噪声图像去噪滤波，去噪后进行二值化处理，通过不断调整二值化阈值得到相对较好的去噪数字字符黑白二值图像，并去除了小于 65 的像素块，以便接下来切割识别。

高斯滤波算法的二维低通滤波器原理：

高斯滤波是一种线性平滑滤波器，它可以通过对图像进行卷积操作来实现图像的平滑效果。原理基于高斯函数（也称为正态分布），其具有以下特点：高斯函数是一个钟形曲线，呈现对称分布，具有一个峰值。高斯函数的形状由其标准差决定，标准差越大，曲线越宽，平滑效果越明显。

高斯函数的二维形式为：$$G(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}$$

具体而言，滤波器输出像素的计算公式为：

$$I'(x,y)=\sum_{i=-N}^{N}\sum_{j=-N}^{N}I(x+i-N,y+j-N)\cdot H(i,j)$$

定义一个二维的高斯核或滤波器模板，该模板是一个权重矩阵，大小为（2N+1）×（2N+1），其中 N 是指定的滤波器的半径。

在图像上滑动滤波器模板，将模板的中心点与图像的每个像素对齐。

对于每个像素，将滤波器模板与图像的局部区域进行逐元素相乘，然后将乘积相加，得到输出像素的值。

重复步骤 2 和步骤 3，直到处理完整个图像的所有像素。

在每个像素位置上，滤波器模板中的每个元素与其对应的图像区域中的像素进行逐元素相乘，然后将乘积相加，得到输出像素的值。乘法的目的是根据滤波器模板的权重对像素进行加权，相加的目的是将加权的像素值进行求和。最终输出的像素值代表了该位置的像素与其周围像素的加权平均值。

高斯滤波器的权重由高斯函数确定，根据像素与中心像素的距离计算。高斯函数是一个钟形曲线，其形状由标准差（sigma）决定。距离中心像素越远的像素获得的权重越小，距离越近的像素获得的权重越大。这样可以保持图像的边缘和细节信息，同时实现图像的平滑效果。

二值化原理：二值化是一种图像处理技术，将灰度图像转换为二值图像（只包含两种像素值，通常为黑色和白色）。其原理是根据像素的灰度值与一个阈值的比较，将灰度值高于阈值的像素设为白色（或前景色），灰度值低于阈值的像素设为黑色（或背景色）。二值化的基本原理和步骤：

确定阈值：首先需要确定一个阈值，它是将灰度图像转换为二值图像的分界点。阈值可以根据图像的特性、应用需求或自动算法来确定。

遍历像素：对图像中的每个像素进行遍历。

阈值比较：将每个像素的灰度值与阈值进行比较。

分类设置像素值：如果像素的灰度值大于阈值，则将该像素设置为白色（或前景色）；如果像素的灰度值小于等于阈值，则将该像素设置为黑色（或背景色）。

重复步骤 2 到步骤 4，直到处理完整个图像的所有像素。

通过上述步骤，可以将灰度图像转换为二值图像，其中只包含两种像素值。二值化可以实现对图像中的目标对象与背景的分离，简化图像的表示和处理，便于进行后续的形状分析、边缘检测、目标识别等图像处理任务。

在我的代码中，我使用 matlab 的集成二值化函数 imbinarize 来进行对去噪后的图片进行二值化，也可以通过 graythresh 函数自动确定二值化的阈值，但我使用 0.75 作为二值化阈值。并通过 bwareaopen 函数删除了小于 65 大小的像素块。

1.4.3 目标定位与分割（训练样本切块）

我使用检测连通域的方法，通过算法提取每个连通域（白色像素数字字符区域）的左右上下边界位置，并将其切割为单独的图像。

连通域分割是图像处理中的一种常用技术，用于将图像中具有相同像素值且相互连接的像素区域划分为不同的连通域。其原理是通过像素之间的连接关系来识别和分割出具有相似特征的区域。连通域分割的基本原理和步骤：

确定像素邻域：对于每个像素，确定其相邻的像素，通常是在其周围的 8 个邻域内进行考虑。

遍历像素：对图像中的每个像素进行遍历。

像素标记：对于未标记的像素，将其作为一个新的连通域，并标记为当前的连通域编号。

连通域扩展：从当前的像素出发，通过比较其相邻像素的值与当前像素的值，判断是否属于同一个连通域。如果相邻像素属于同一连通域且未被标记，则将其标记为当前的连通域编号，并将其加入到待处理的像素列表中。

我在 matlab 中使用 regionprops 函数来检查连通域，虽然我自己编写了代码也可实现但效果不如 regionprops 函数好，再按照检测到连通域的左右上下边界对字符进行切割保存。

1.4.4 图像特征提取（KNN 模型训练）

我先将切割好的图像文件的文件名重命名为“图像的数字_序号”形式，再将每个图像按比例调整为 28*28 像素大小，再提取出每个图像的像素值（即图像每个像素的值，黑为 0，白为 1）为二维矩阵，再将每个图像的数字（标签）与二维像素值矩阵一一对应联系起来。

图像特征提取原理：是从图像中提取出能够表示图像内容的有效信息。这些信息通常以数值或向量的形式表示

1.4.5 分类器设计（KNN 模型训练）

将图像的像素值转换为特征向量。对于灰度图像，可以将图像展平为一个一维向量，特征 z 类。KNN 算法使用距离度量来计算实例之间的相似性。常见的距离度量方法包括欧氏距离、曼哈顿距离、闵可夫斯基距离等。根据具体问题和数据特征，选择合适的距离度量方法。对于要预测的新数据样本，KNN 算法计算它与训练集中每个样本之间的距离，并选择与其最近的 K 个样本作为其邻居。对于分类问题，KNN 算法通过投票表决来确定新数据样本所属的类别。即，根据邻居中的多数类别标签来判断新样本的类别。在回归问题中，可以使用邻居样本的平均值或加权平均值作为预测值。

在我的代码中，我使用 fitcknn(trainX_fold, trainY_fold, 'NumNeighbors', 10);函数来搭建 knn 分类器，trainX_fold 为图像特征矩阵，trainY_fold 为图像标签，fitcknn 默认使用欧式认使用欧氏距离作为距离度量方法。最后得到训练好的模型。

1.4.6 手写数字识别

点击“手写数字识别”按钮，选择一个 bmp 图像读取待识别手写数字图像，对其进行降噪二值化调整大小处理，提取其特征与训练好的模型比对、得到识别结果。并在程序中显示识别结果。

1.5 用户界面

我基于 matlab R2023a 的 app designer 功能，设计了基本用户交互窗口，将上述功能分别置于按钮中，用户按下对应功能按钮即可运行模块，还会将输出信息置于交互界面消息框中。

2.系统使用说明及演示

进入系统即可看到初始交互界面，如图 2.1 所示。

图 2.1 行驶证号牌号码识别系统初始界面

点击“训练样本降噪处理”，即可在弹出的文件选择框中选择需要训练的样本文件夹与输出文件夹，如图 2.2、2.3 所示。

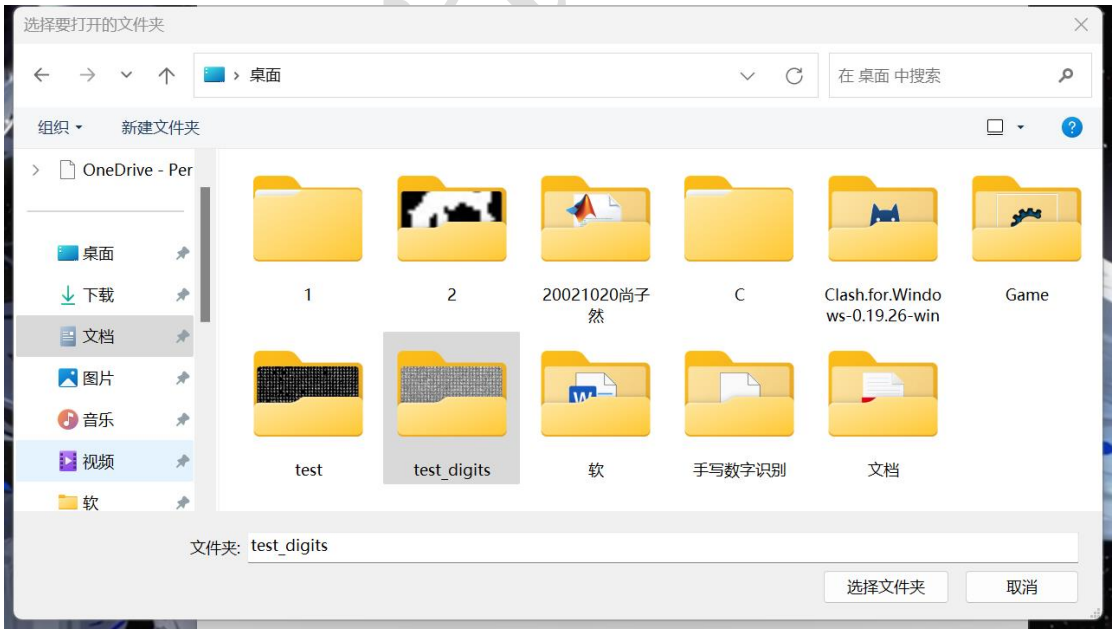

图 2.2 选择需要训练的样本文件夹界面

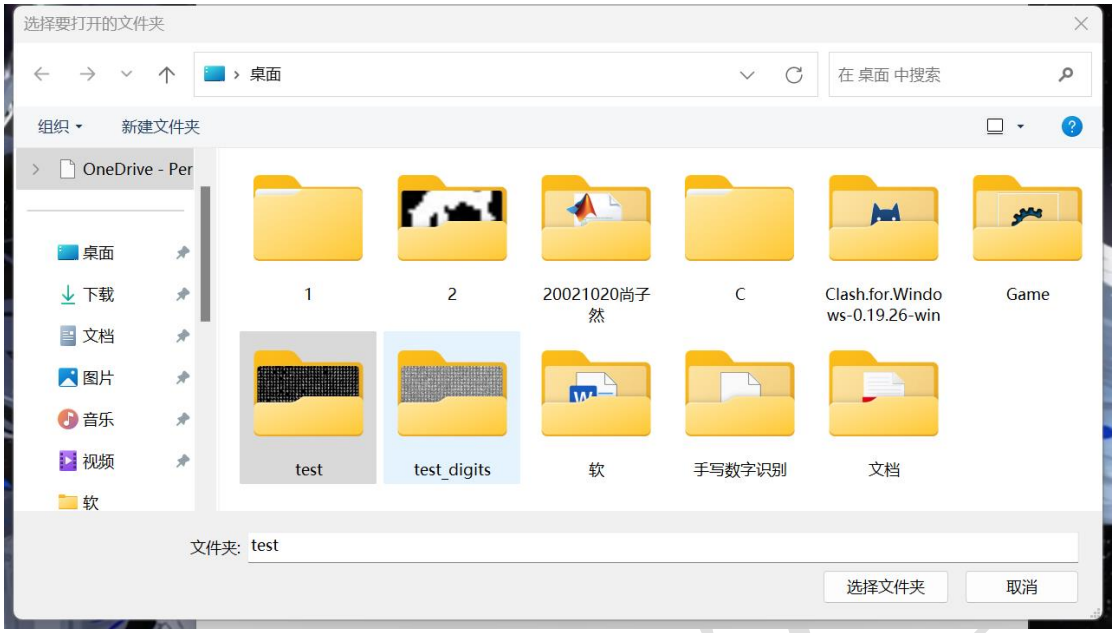

图 2.3 选择输出文件夹界面

训练样本为加噪后的 minst 手写数据集，如图 2.4

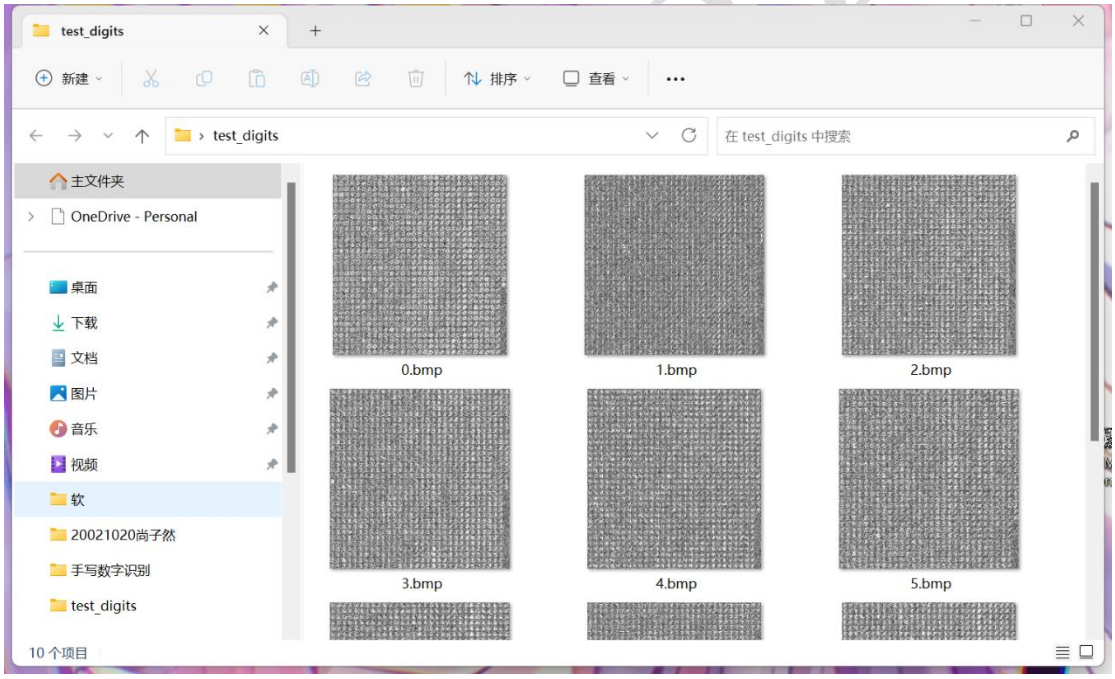

图 2.4 加噪后的 minst 手写数据集

选择完毕输入输出文件夹后，系统开始进行训练样本降噪处理。此时消息框上会提示信息已开始运行，如图 2.5 所示。

图 2.5 开始运行界面

运行完成后，也会提示消息，如图所示。

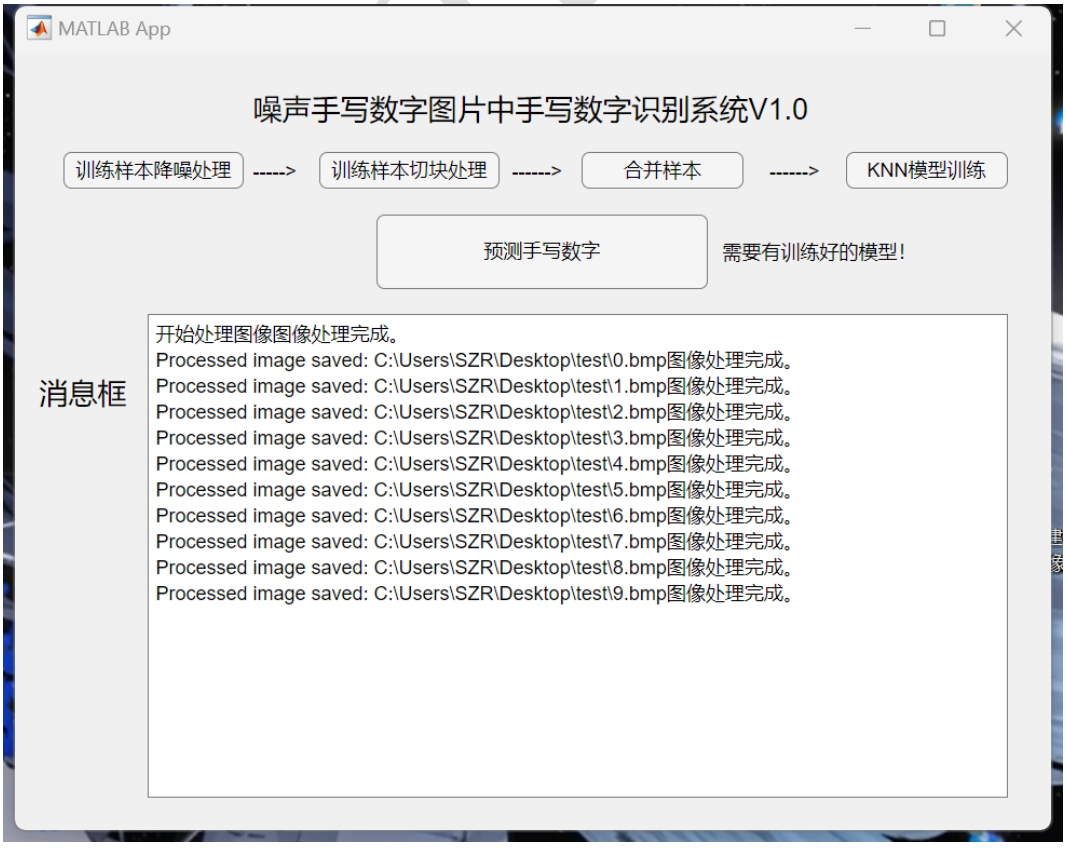

图 2.6 运行完成显示界面

点击“训练样本切块处理“按钮，即可在弹出的文件选择框中选择需要处理的样本文件夹与输出文件夹，应选择上一步处理完成的输出文件夹，如图 2.7、2.8、2.9 所示。

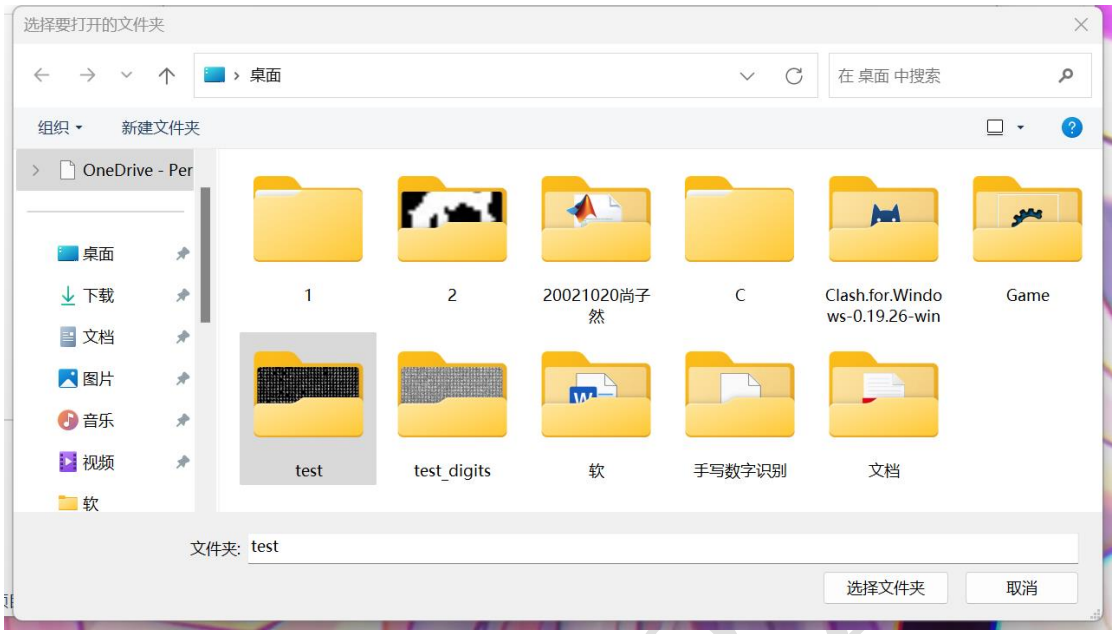

图 2.7 选择上一步处理后文件路径界面

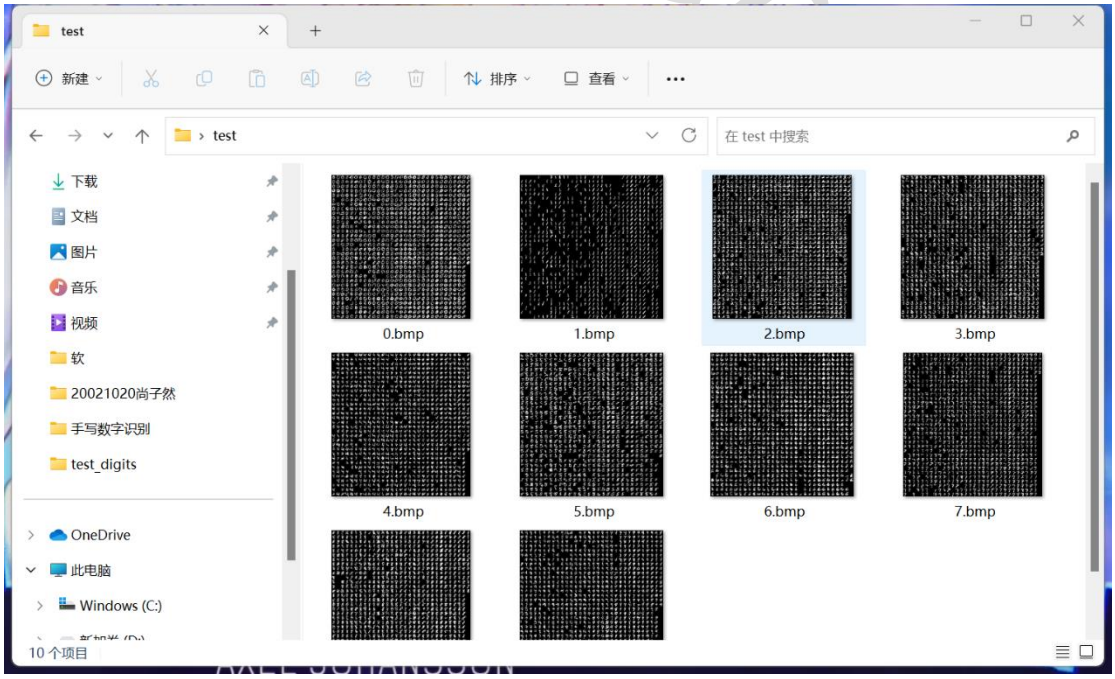

图 2.8 选择上一步处理后文件夹内容

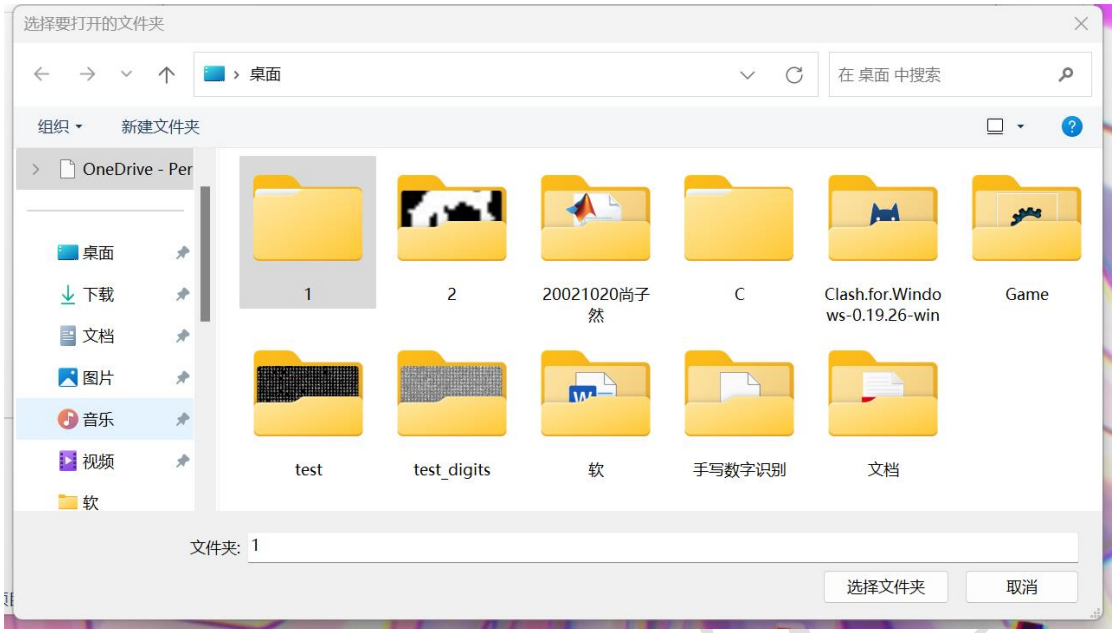

图 2.9 选择保存输出文件路径界面

选择完成后、程序开始执行训练样本切块处理，消息框会如图 2.10 提示.

图 2.10 开始执行训练样本切块处理

运行完成后消息框会提示如图 2.11 消息。

图 2.11　运行完成

运行完成后样本如图 2.12、2.13 所示。

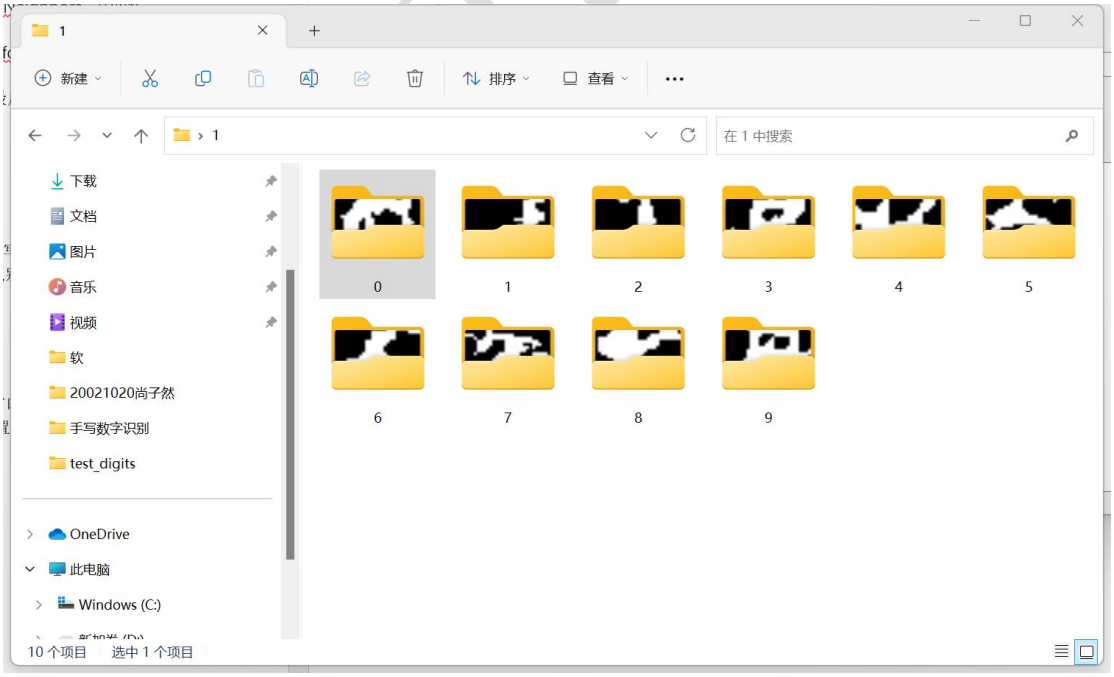

图 2.12 运行完成训练样本文件夹

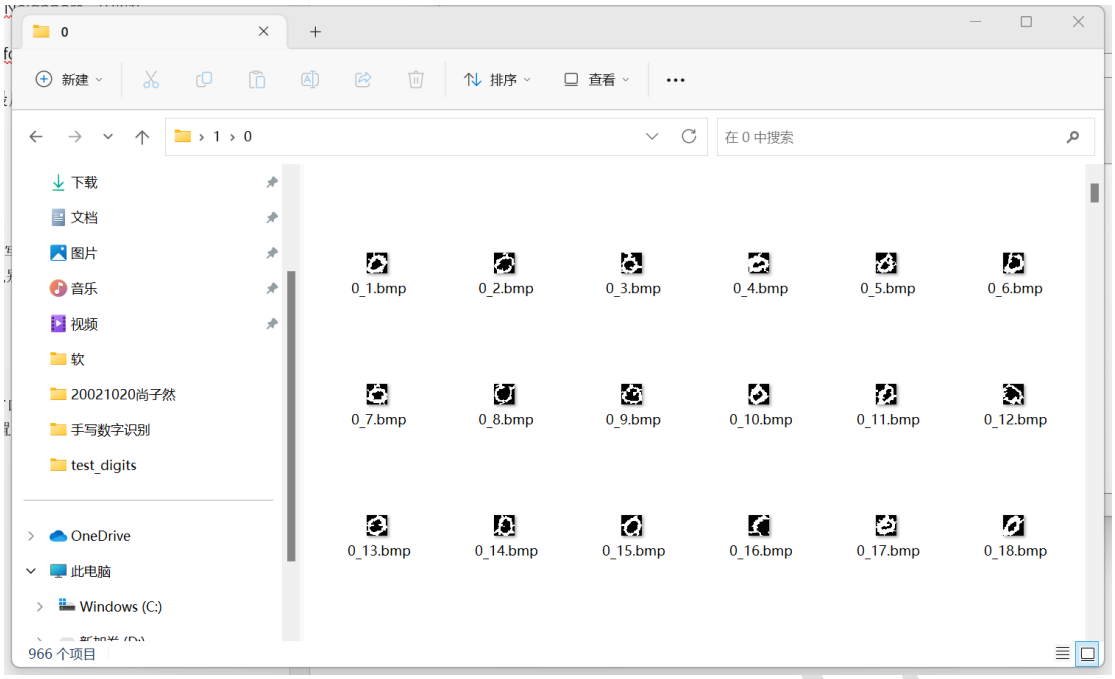

图 2.12 运行完成训练样本

点击“合并样本 “按钮，即可在弹出的文件选择框中选择需要处理的样本文件夹与输出文件夹，应选择上一步处理完成的输出文件夹，选择过程与上几步相同，可参照图 2.7、2.9。程序开始运行，提示消息如图 2.13.

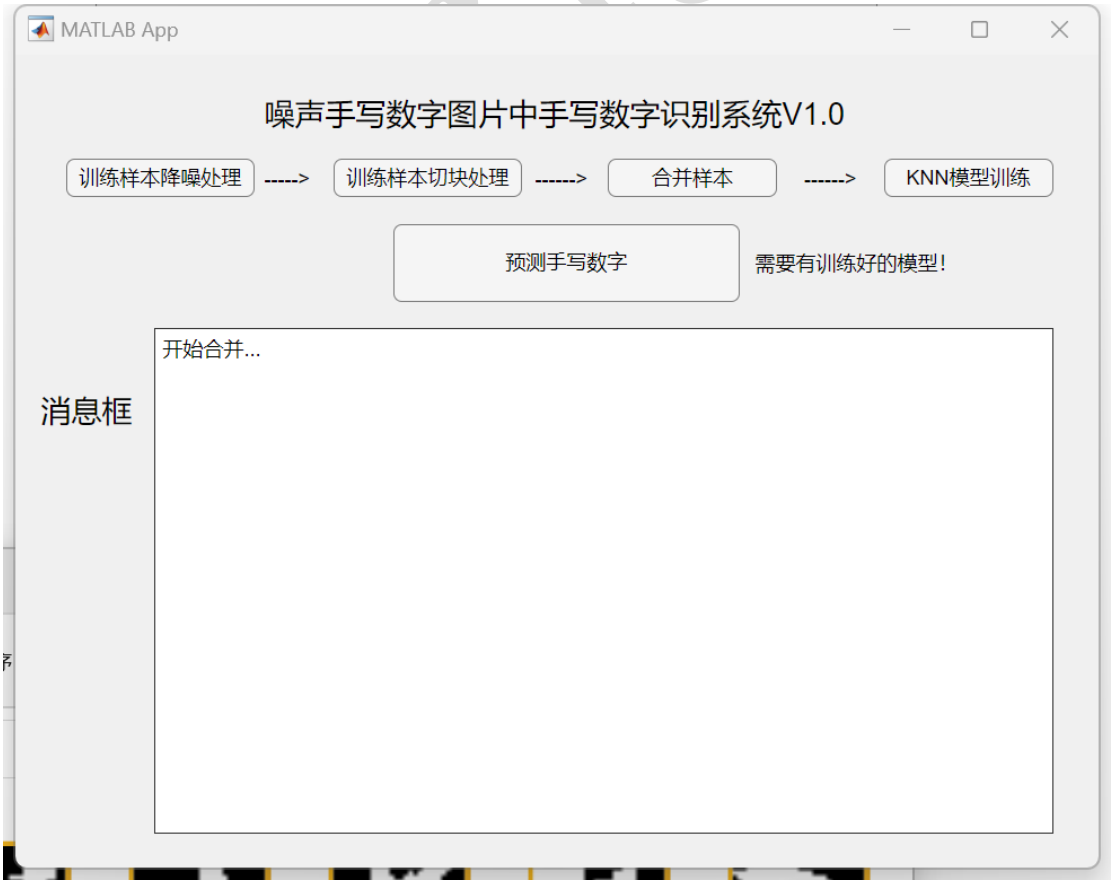

图 2.13. 开始运行提示消息

运行完成后提示消息如图 2.14.

图 2.14　运行完成后提示消息

点击“KNN 模型训练”按钮，即可在弹出的文件选择框中选择需要处理的样本文件夹，应选择上一步处理完成的输出文件夹，如图 2.15 所示。

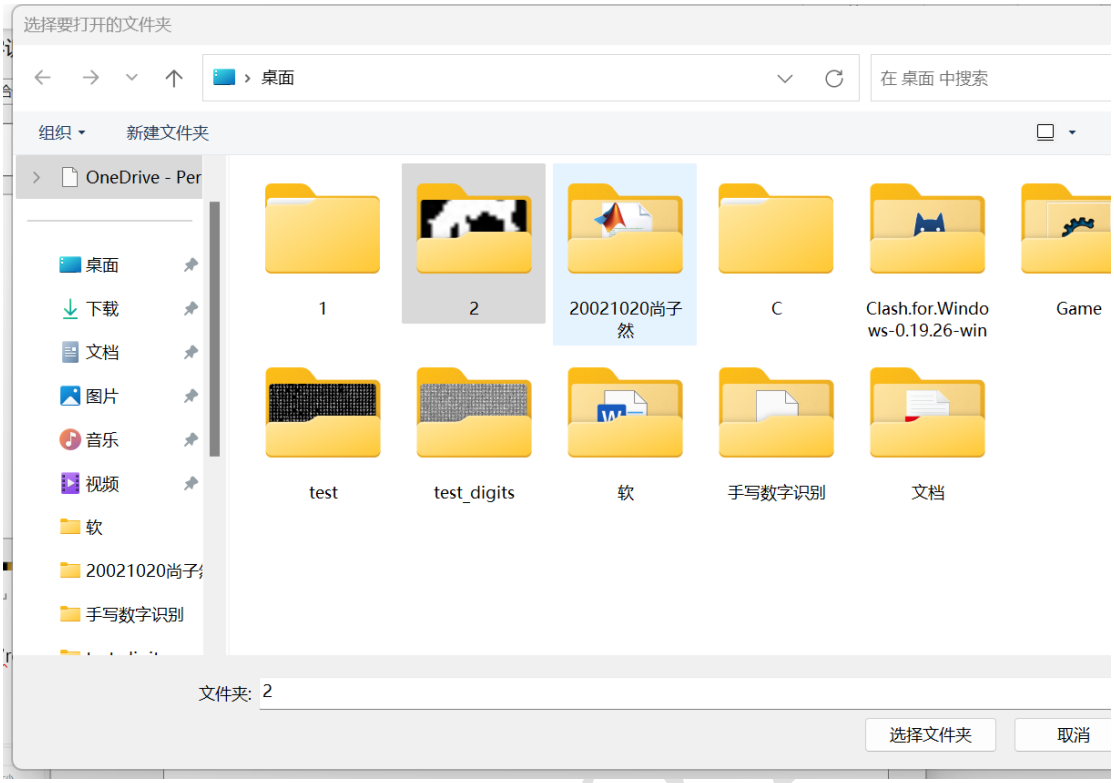

图 2.15 选择上一步处理完成的输出文件夹

选择完成，系统开始执行 KNN 模型训练，消息框提示如图 2.16 所示。

图 2.16 消息框提示开始训练

训练完成后，系统自动保存训练好的模型与模型预测混淆概率矩阵到同一路径下，如图所示。

图 2.17 训练完成后提示消息

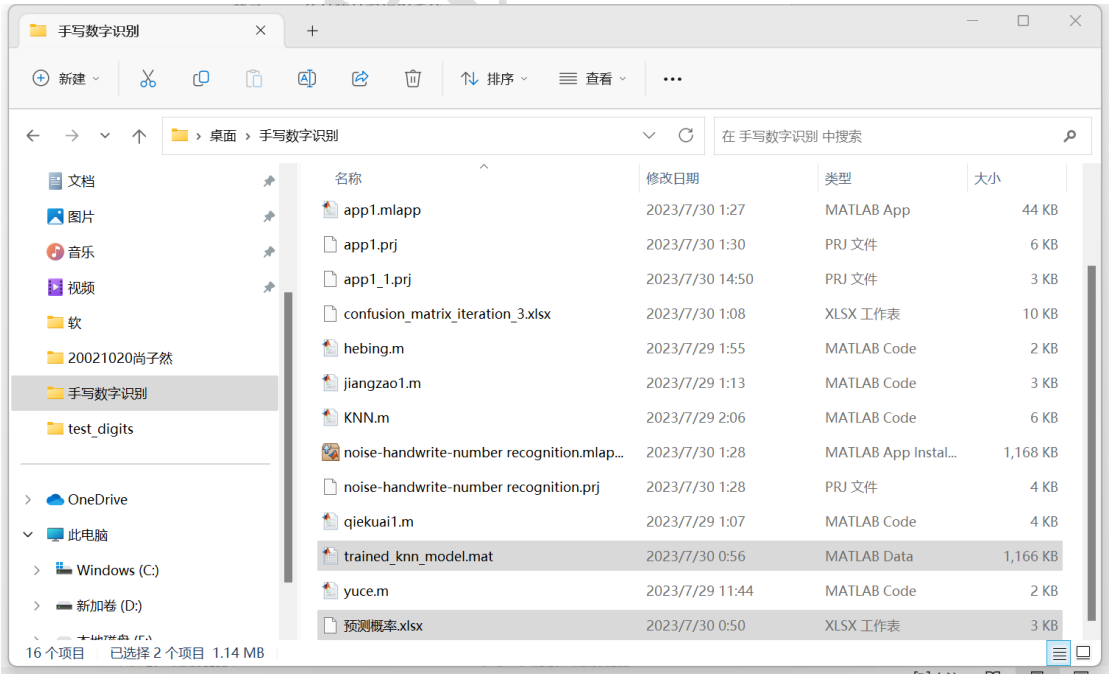

图 2.18 保存的文件

点击“预测手写数字”按钮，系统将读取同一路径下的训练好的模型文件，即可在弹出的

文件选择框中选择需要识别的手写数字 bmp 图像文件。如图 2.19 所示。

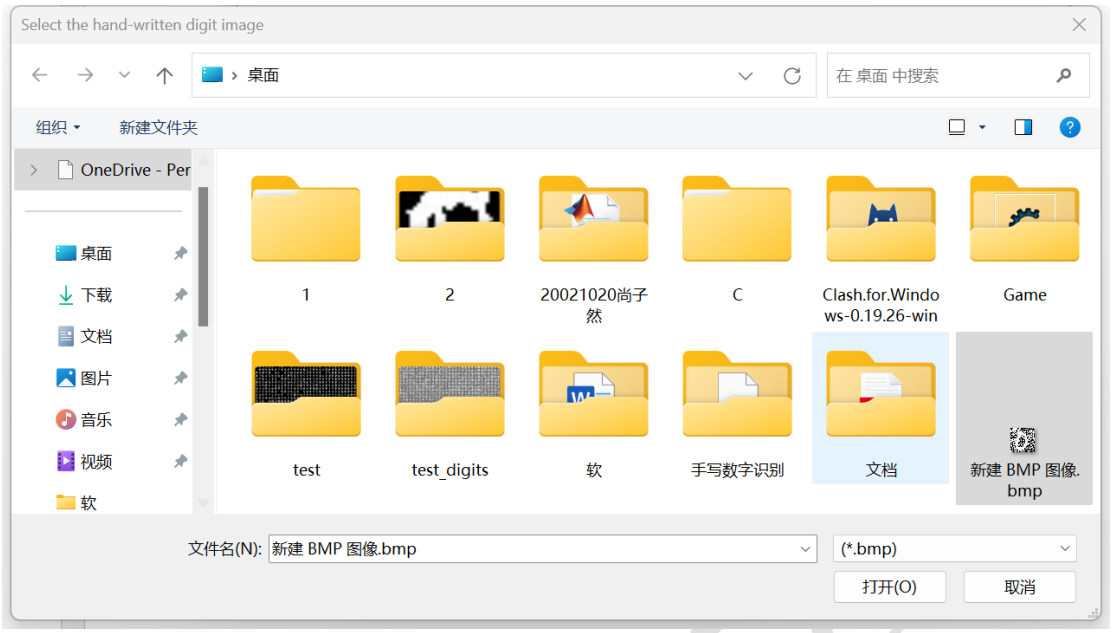

图 2.19 择需要识别的手写数字 bmp 图像文件

选择完成后，系统开始处理图像并给出处理过程，最后在消息框给出识别结果。如图 2.20 所示。

图 2.20 处理过程与出识别结果显示