**基于LSTM的在线手写体汉字识别**

1、项目简介

触摸式用户接口的日益普及使得其在人机交互领域发挥重要作用。采用手指或笔进行输入的平板电脑、智能手机和TUI（触摸式用户接口）电脑正在成为人们生活中不可或缺的一部分。作为重要的数据输入设备，手指或笔已经代替了传统的鼠标和键盘的许多功能，因此，如何对通过手指或笔等输入设备书写的汉字进行在线实时识别成为了当前热点的研究方向。不同于利用键盘等具有明确意义的输入设备，手指或笔作为输入设备时会受到用户个体书写习惯的影响，因此即使对于同一个字，设备也有可能识别出不同的输出结果。为了明确得到用户的输入意图，准确识别输入文字具有重要的现实意义。手写字体识别通过对不规则的手写字体进行特殊的处理，从而可以被应用于智能阅卷、自动识别学生的作业和试卷的手写内容，从而可以促进教学管理的数字化和智能化。此外，手写字体识别还可以通过自动识别手写的图书摘要、读数笔记等，提高阅读效率。最后，通过识别电子手写表单的内容，通过对纸质形式的信息进行统计整理，以降低人工录入的成本，方便登记相关信息。

当前手写字体识别的主要困难包括以下两个方面，首先，由于汉字字符数量较大，即使人们日常使用的汉字也有高达三千多个，这三千多汉字中还不包含日常生活中不常使用的汉字，因此，对数量庞大的汉字进行准确的识别具有很大的难度。其次，中文汉字字体种类繁多，并且不同字体之间的字型具有很大的相似性，当人们在书写汉字的过程中产生书写不规范现象时，增加了准确识别字体的难度。此外，中文汉字繁多，包括楷书、行书、草书、小篆等不同的字体，而不同字体之间的同一个汉字也具有不同的形态，因此能够准确识别不同字体之间的汉字也是十分具有难度的工作。因此，项目基于常见的3000多个汉字，实现手写字体的识别工作。

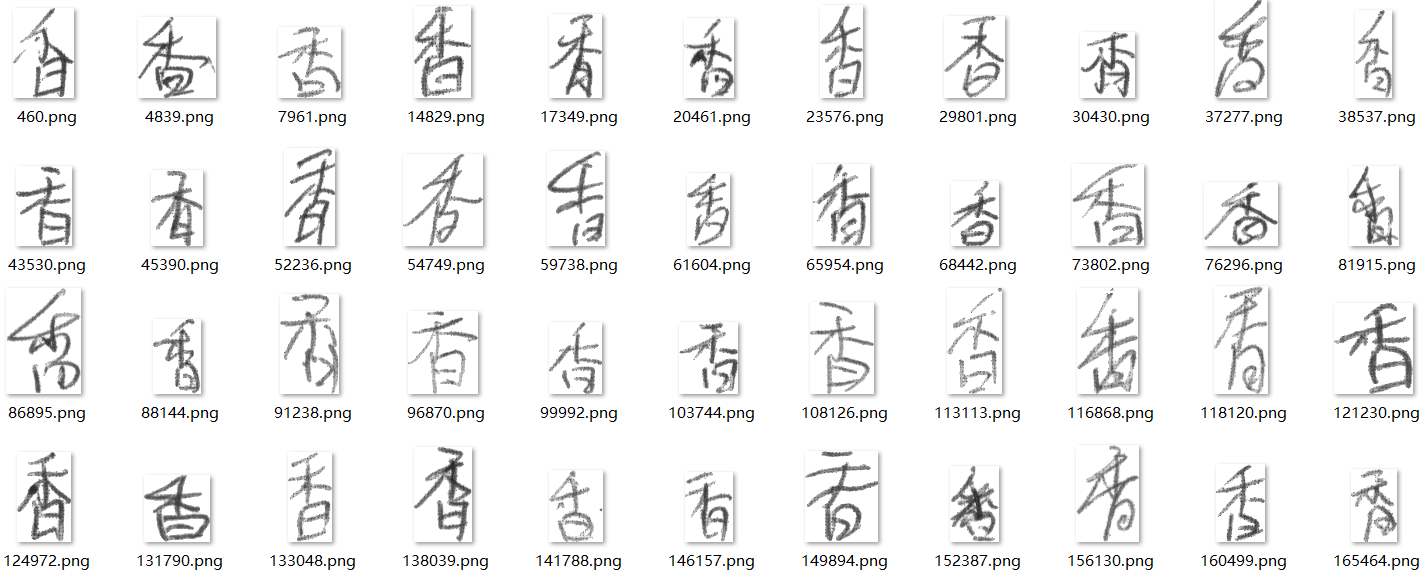
2、本文所用LSTM算法原理

2.1 在线手写体汉字识别数据集

数据集的质量和数量会影响模型的泛化性和在线手写体汉字识别的准确率，为了能够准确的识别到手写的汉字，可以通过爬虫和寻找公开数据集的方式进行数据收集。通过调研发现，已有开源数据集包括CASIA-HWDB、CASIA-OLHWDB、NIST等。考虑到本文是对手写体汉字进行识别，因此选择中科院自动化所开源的HWDB作为数据集进行模型的训练。CASIA-HWDB和CASIA-OLHWDB数据集是由中科院自动化所于2007年到2010年耗时3年时间所收集得到，该数据集包含有1020名用户联机和手写两种方式的中文单字体样本，通过利用专业设备在点阵纸上进行书写扫描和字体分割后所得到。CASIA-HWDB-1手写单字体识别库包含有3个子版本，共3895135个手写单字样本，该样本被划分为7356类，7356类包含有7185个汉字和171个英文字母、数字和符号。不同于CASIA-HWDB-1，CASIA-HWDB-2版本共有5091页图像，每张图像上包含有多行文本，对于多行文本，研究人员将5091页图像分割为52230个文本行和1349414个单一文字，对于所有的文字均祛除了背景信息并被转化为灰度图像，最后按照用户书写的顺序进行存储，部分数据集中的汉字如图3-1所示。



(a) “骄”



(b) “香”



(c) “饼”



(d) “颤”

图3-1 HWDB数据集示例图像

Figure 3-1 An Example of HWDB Images

对原始数据集进行分析可以发现，原始数据集中图像大小不同，因此，需要在进行特征提取之前需要首先对原始图像进行归一化处理。对图像大小进行分析后本文将不同大小的图像统一归一化为宽和高均为64大小的图像。

3.2 本文所用LSTM算法模型

通过对相关文献进行调研可以发现常用的手写汉字识别框架包括CNN+LSTM+CTC（CRNN+CTC）和CNN+Seq2Seq+Attention两种方法，本文基于第一种方法实现手写汉字的识别工作。CRNN是一种卷积循环神经网络结构，用于解决基于图像的序列识别问题，特别是文字识别问题，通过卷积神经网络将图片的特征出来后采用LSTM对序列进行预测，CRNN+CTC算法模型原理如图3-2所示。

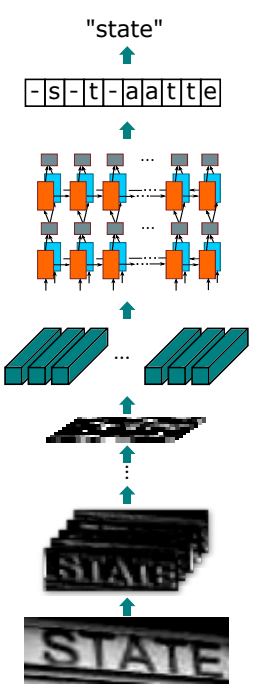


图3-2 CRNN+CTC算法模型原理

Figure 3-2 LSTM model algorithm

CRNN网络结构包含卷积层、循环层和转录层三部分，其中卷积层用于对原始图像进行特征提取，得到原始图像的特征图。循环层使用双向长短期记忆网络LSTM对特征序列进行预测，对序列中的每个特征向量进行学习，并输出预测标签分布。转录层使用CTC损失，将从循环层获取的一系列标签分布转换成最终的标签序列。

3.2.1 卷积层结构图

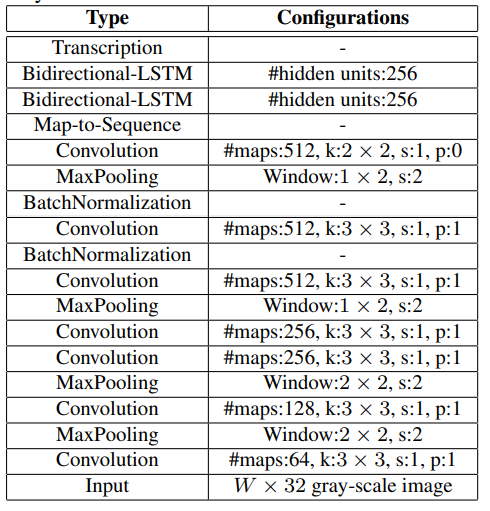


图3-2 卷积层结构

Figure 3-2 Network configuration summary

用于特征提取的卷积层包括四个最大池化层，六个特征提取层和两个归一化层。其中，最后两个池化层对池化窗口的尺寸进行了更改，进行将图像的高度缩小了4倍，而宽度仅缩小了2倍，从而适应文本图像高而小且宽而长的特点，因此可以保证在对原始图像进行特征提取时不丢失图像在宽度上的信息。模型引入归一化模块的目的是加快模型的收敛，缩短训练过程。模型以单通道的灰度图像作为输入。

3.2.1 Map-to-Sequence

对图像进行特征提取后，提取到的特征无法直接用于长短期记忆网络LSTM进行训练，因此需要对特征进行处理，将提取到的特征层处理为模型所需的特征向量序列。在特征图上按照从左到右的顺序生成特征向量，每一列包含有512维的特征向量，因此，第i个特征向量是所有特征图第i列像素的连接，特征向量便构成了序列。由于卷积层和池化层具有平移和旋转不变性，因此特征图的每一个特征向量对应于原始图像的每一个感受野，并且这些矩形区域与特征图上从左到右的相对应列具有相同的顺序，特征序列中的每个向量关联一个感受野。Map-to-Sequence结构如图3-3所示。

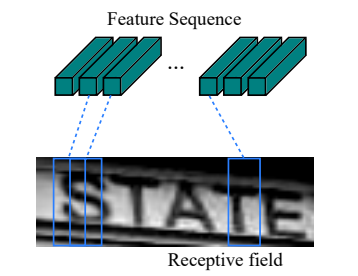


图3-2 视觉感受野

Figure 3-2 The receptive field

3.2.2 长短期记忆网络

在训练传统的循环神经网络RNN模型时会出现梯度消失的问题，梯度消失会影响获取到上下文特征，因此本文利用长短期记忆网络LSTM模型作为序列预测模型。长短期记忆网络LSTM结构的特殊性使得其可以捕获长距离的依赖，但是由于原始长短期记忆网络LSTM为单向模型，仅会利用过去的信息，然而在图像序列中，正向和反向上下文信息具有互补性优势，因此本文利用两个长短期记忆网络LSTM，分别提取图像的前向特征和后项特征，形成双向LSTM，此外，可以通过堆叠多个双向LSTM增加模型的深度，从而可以提取到更深层次的抽象特征。LSTM结构如图3-3所示。

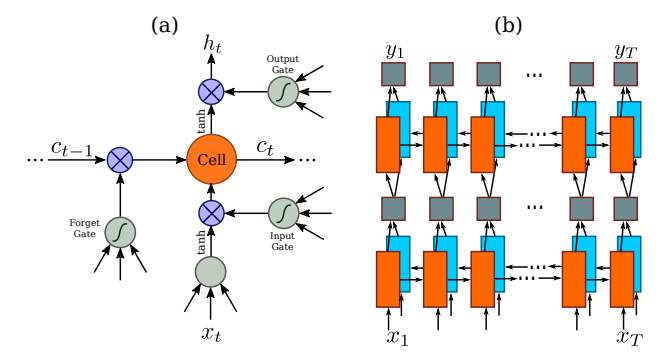


图3-3 LSTM模型结构

Figure 3-2 The Structure of LSTM

3.2.3 CTC损失

得到序列特征后需要将LSTM对每个特征向量所做的预测转换成标签序列。因此需要根据每帧预测得到具有最高概率组合的标签序列。采用端到端的方式进行手写汉字识别需要利用CTC损失实现CNN与LSTM进行联合训练。LSTM进行时序分类时，会出现冗余信息，因此需要去除掉冗余信息。在CTC中，通过blank机制来解决冗余信息，LSTM输出序列时，通过在文本标签中的重复字符之间增加特殊符号的方式减少冗余信息，当特殊符号间存在字符隔开时，连续相同字符便不进行合并。

4、基于LSTM的在线手写体汉字识别模型测试及系统设计

4.1 实验数据集及实验环境

4.1.1 环境配置

本文利用Anaconda集成平台，基于Python语言进行程序的开发，Python语言是开源语言，生态完善，对于多数深度学习框架都提供了调用接口，用户可以在任何计算机上免费安装和使用相关数据库。此外，Python语言易于学习，语法简单，代码维护简单，拥有丰富的扩展库，可以轻易的完成各种高级任务。

首先需要在Anaconda官网下载Anaconda软件进行安装，并进行环境变量的配置。安装结果如图4-1所示。软件安装完毕后，新建虚拟环境，并进行python 3.8和相关依赖库的安装，相关依赖库包括Pytorch、opencv-python、tensorflow深度学习框架等。

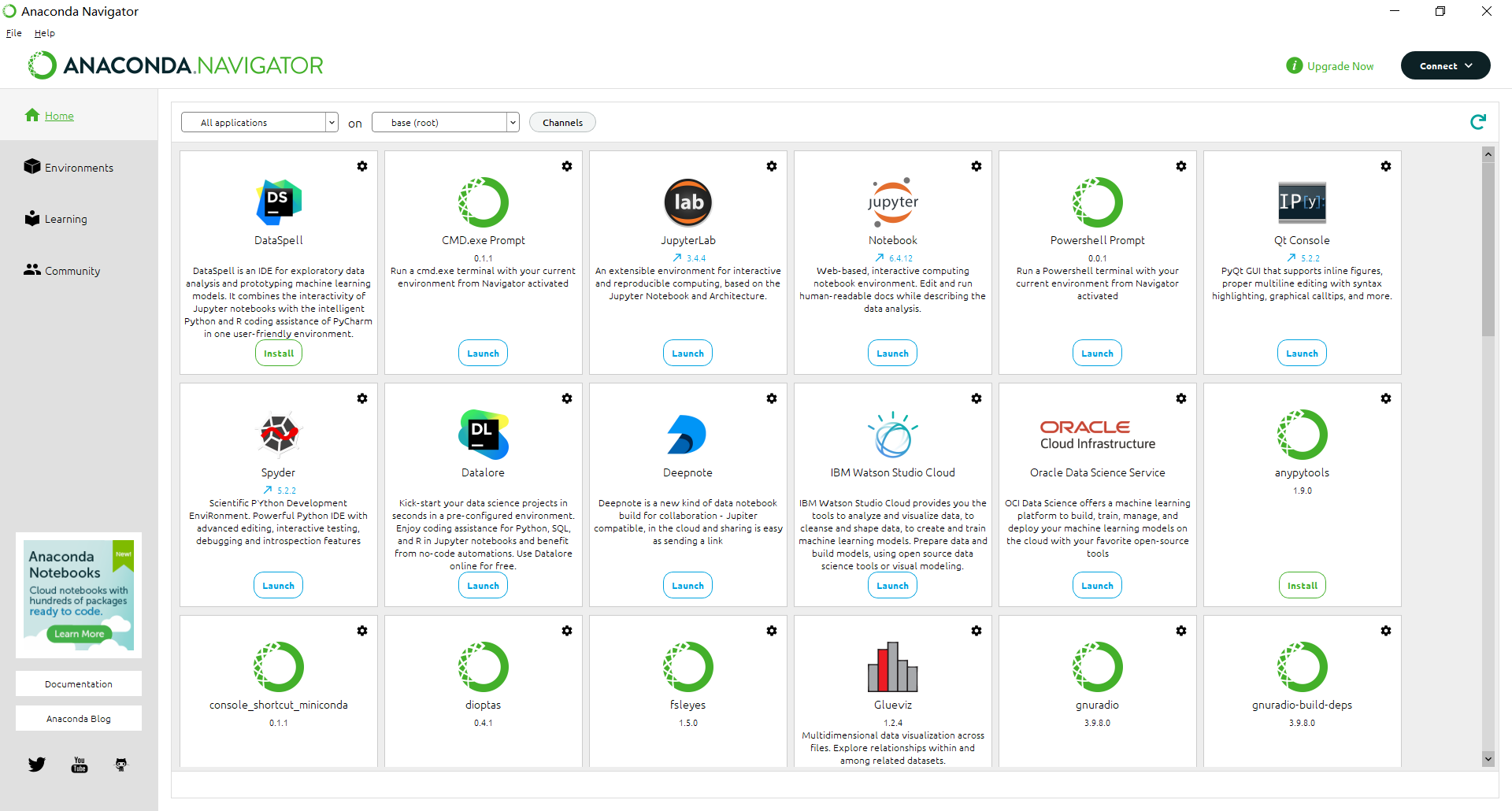


图4-1 软件安装结果

Figure 4-1 Anaconda install

本文开发的实验环境为Windows系统，GPU型号为Nvidia 2080，为了能够充分的对模型进行训练，模型训练轮次为20万轮。

4.1.2 实验数据集

利用第三章所示收集的在线手写体汉字数据对模型进行训练。将所有数据均分为训练集和验证集，数据集组织形式如图4-2所示。

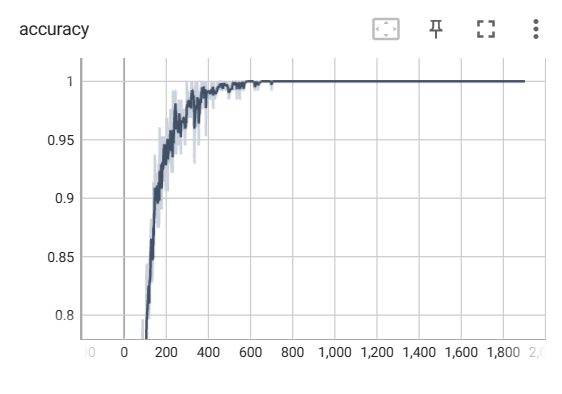


图4-2 数据集组织形式

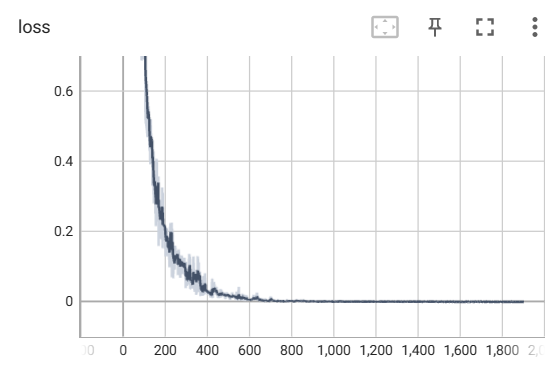
Figure 4-2 The structure of dataset

每一个数据集中包含有同一个汉字的不同书写方式，文件夹名称为训练的标签，数据集中包含有常见的3755个汉字。每个文件夹中的图像个数不同，为了防止模型中出现类别偏差，对不同文件夹中的数据进行分析可以发现，不同类别文件夹中的图像数量相似，因此不会出现模型偏差。为了提高模型的鲁棒性和泛化性，在模型的训练过程中应用了数据增强策略，数据增强策略包括对原始图像进行旋转、添加高斯噪声、图像归一化等。

在模型训练过程中保存模型训练的评估指标，评估指标包括训练过程中的准确率、损失值和测试过程中的的准确率和损失值。此外，为了防止模型过拟合，在模型训练过程中每隔1000轮保存一次模型，模型保存策略为当该次训练精度高于上次训练精度时进行模型保存，如果该次训练精度不高于上次训练精度，则不保存该模型。模型训练准确率和损失值结果如图4-3（a）和图4-3（b）所示。结果显示，当模型训练轮次达到6万轮之后，模型准确率不再有较大的上升趋势，同样，通过分析损失函数曲线也可以发现该区域，当模型训练轮次达到6万轮之后，模型的损失函数不会发生较大的下降，但是由于防止训练不充分对准确率的影响，我们仍然将模型训练了20万轮，此时会出现过拟合现象，过拟合现象即模型仅在训练集上拥有较高的识别准确率，但是在测试集上模型准确率下降。为了防止过拟合现象，我们在实际应用过程中利用第6万轮保存的模型进行实际和测试集的测试，将6万轮之后的模型舍弃。



1. 模型训练准确率

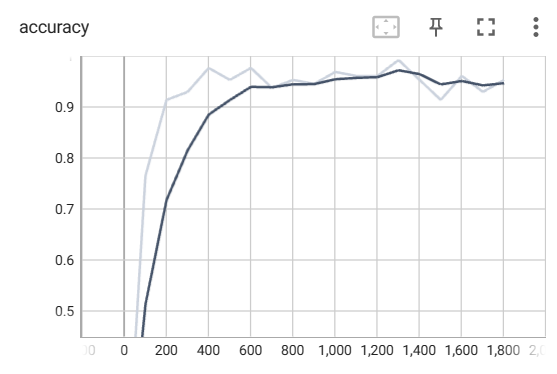


1. 模型训练损失

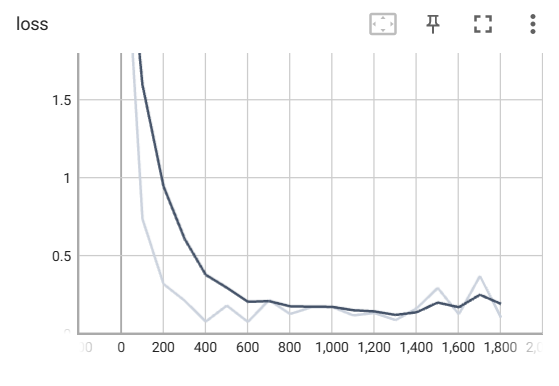
图4-3 模型训练过程

Figure 4-3 Train Process

模型测试准确率和损失结果如图4-4（a）和图4-4（b）所示。测试准确率和损失与训练准确率和损失具有相似的特点，因此本文选择第6万轮的模型进行测试可以很好的满足泛化性。



1. 模型测试准确率



1. 模型测试损失

图4-4 模型测试过程

Figure 4-4 Test Process

为了进一步验证模型的泛化性和鲁棒性，基于第6万轮保存的模型，利用测试集对模型进行测试，统计模型Top1和Top3的准确性来测试模型的性能，其中Top1代表模型准确预测到输入汉字的概率，输出的汉字与输入的汉字完全匹配。Top3代表模型对输入预测出三个汉字，三个汉字中如果包含有输入的汉字，不考虑汉字在预测结果中的顺序，此时表明模型预测结果准确，测试结果如图4-5所示。结果显示在测试集上，模型Top1的测试准确率为92.49%，模型Top3的准确率为97.47%。

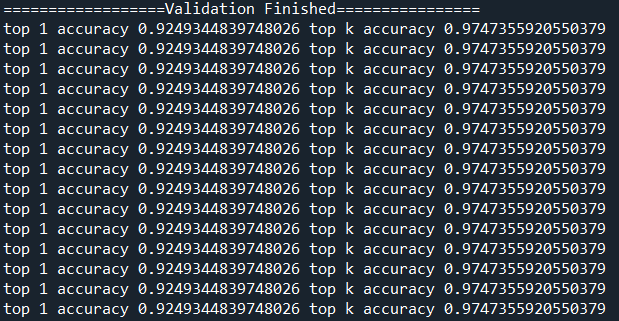


图4-5 模型Top-1和Top3结果

Figure 4-5 Results of Top1 and Top3

接下来我们通过可视化的形式进一步验证模型的准确率。通过如图4-6所示形式的图片作为输入验证模型的预测效果。每张测试图片上包含有每名用户书写的多个大小不一、形态各异的汉字。

test1test2

图4-6 部分测试数据

Figure 4-6 Part of test data

实验结果如图4-7所示，其中results为模型输出结果，gt为真实结果。实验结果显示模型可以很好的预测序列输入图像上的汉字。在序列图像上预测到的汉字的平均准确率可达83%。

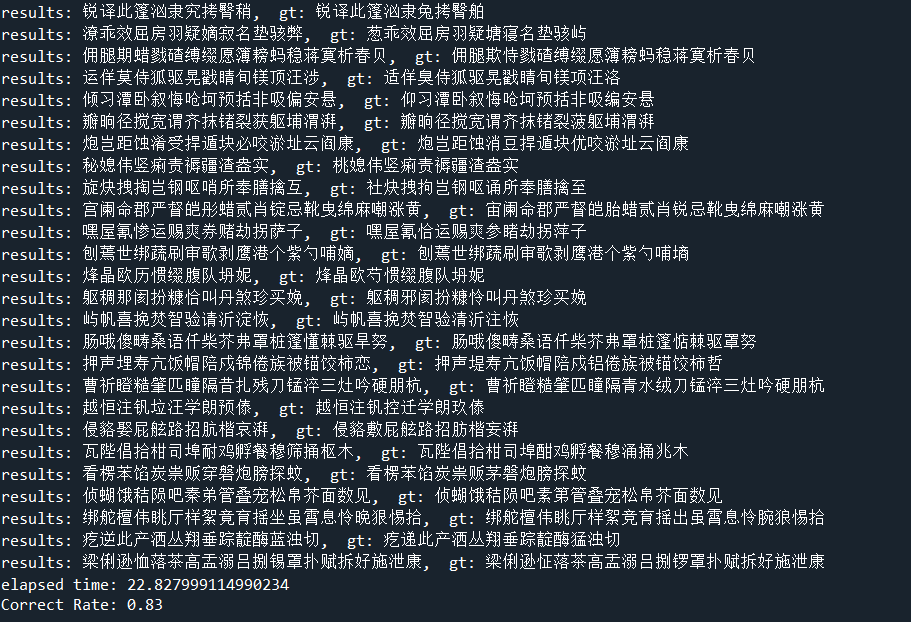


图4-7 部分测试结果

Figure 4-7 Parts of test results

4.2 界面设计实现及测试

通过定量的对模型进行测试后，为了能够实现对实时输入的手写体汉字进行识别的功能，我们设计了如图4-8所示的交互界面，交互界面包括使用说明，模式选择，数据清除，识别和数据输入5大部分。使用说明部分详细对该系统的功能进行了介绍，并对4种识别模式和使用方法进行了说明。手写输入汉字识别系统包含有4种模式，第一种模式为数字随机抽取，第二种模式为数字手写输入，第三种模式和第四种模型分别为两种不同的LSTM模型用于测试手写汉字输入的识别准确率，其中第三种模型基于卷积神经网络，第四种方式将卷积神经网络和LSTM相结合实现手写汉字的识别。数字随机抽取即随机抽取测试集中的数字，对抽取到的数字进行识别。数字手写输入即在数据输入区域通过鼠标书写待识别的数字，然后对该数字进行识别。汉字随机输入即在数据输入区域通过鼠标书写待识别的汉字。

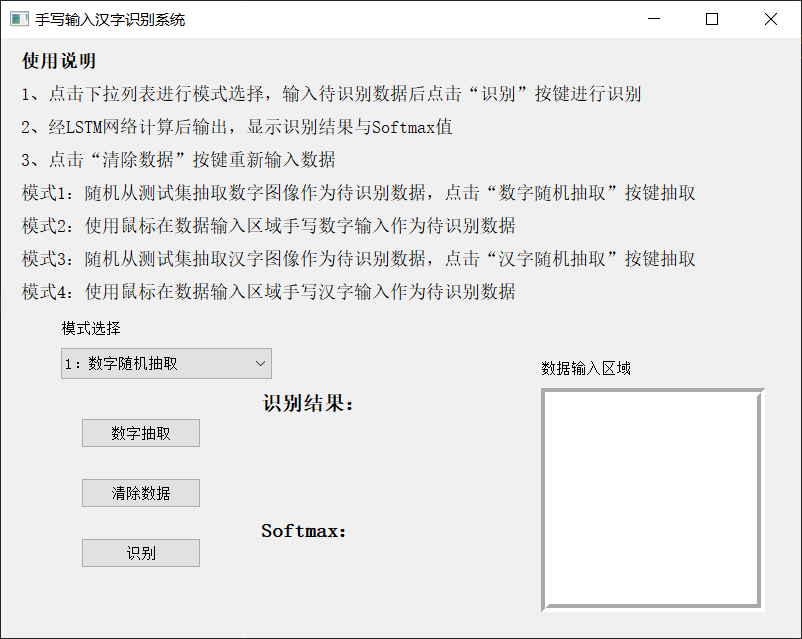


图4-8 基于LSTM的在线手写体汉字识别交互系统

Figure 4-8 The interaction system of handwritten recognition based on LSTM

模式选择界面如图4-9所示。

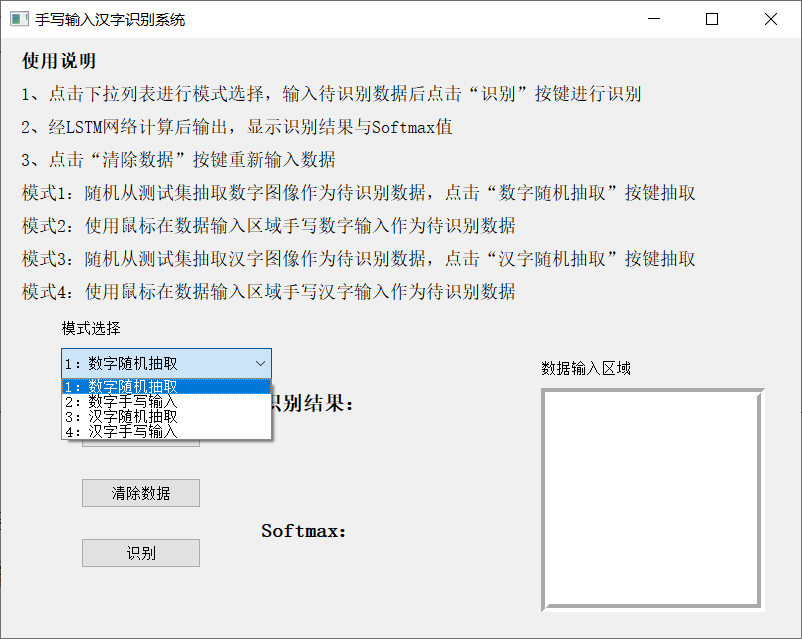


图4-9 识别模式选择

Figure 4-9 Recognition Pattern Selection

数字随机抽取模式下识别结果如图4-10所示。结果显示，对从测试集中抽取出来的标签为7的图像的识别结果为7，识别结果的置信度可到0.9999。在同一模式下进行输入数字识别时，需要点击系统中的清除数据按钮实现数据输入区域内容的清除，清除数据后可点击数字抽取进行新的数据预测。



图4-10 数字随机抽取模式识别结果

Figure 4-10 Random Select Result of Digital Recognition

当模型由数字抽取变为数字手写输入后，此时数字抽取按钮将被锁定。对于数字手写输入，由于利用鼠标进行数字书写时需要在数据输入区域展示书写的内容，因此为了增加对比度，此时输入区域的背景为黑色背景，如图4-11所示。

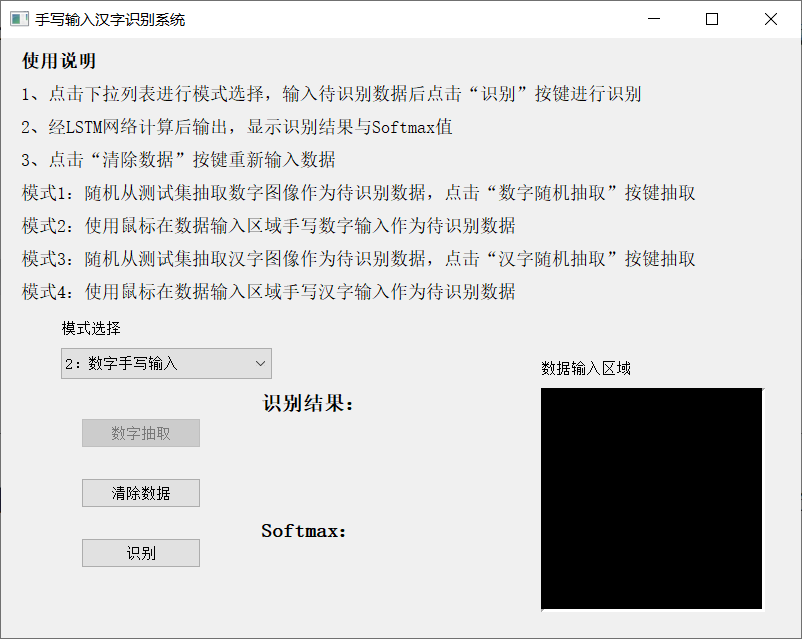


图4-11 手写输入背景转换结果

Figure 4-11 Background Transfer Result

此时在数据输入区域可进行数字的书写，当书写完成后，点击识别按钮实现手写数字的识别，以数字5为例，识别结果和该结果的识别置信度如图4-12所示。



图4-12 手写数字识别结果

Figure 4-12 Digital Recognition Result

汉字手写输入法1模式下的操作方式与手写数字识别模式相同，采用CNN作为特征抽取和识别，抽取按钮将会变为锁定模式，我们随机在数据输入区域利用鼠标书写了一个“好”字，模型的实时测试结果如图所示，可以识别出数据输入区域的“好”字，汉字手写输入法1模式下识别结果如图4-13所示。



图4-13 汉字随机抽取识别结果

Figure 4-13 Random Select Result of Handwriting Recognition

汉字手写输入法2模式与汉字手写输入法1模式相同，不同的是，模型利用CNN+LSTM进行模型的抽取和分类，汉字手写输入法2模式识别结果如图4-14所示。结果显示，对汉字“8”的识别结果为8，结果的置信度为60.73%。



图4-14 汉字手写输入识别结果

Figure 4-14 Handwriting Recognition Result

不同于已有研究工作基于卷积神经网络对图像进行处理，本文将图像数据转化为时序数据，利用数据间的时序关联性进行手写汉字识别，为了进一步测试基于LSTM进行手写汉字识别的效果，接下来将展示更多的识别结果。

4.3 在线手写体汉字结果验证

接下来我们将展示基于LSTM算法实现在线手写体汉字识别的更多测试结果，相关测试结果如图4-7(a)-4-7(e)所示。



(a) “好”结果



(b) “人”结果



(c) “出”结果



(d) “由”结果

图4-7 在线手写体汉字识别测试结果

Figure 4-7 Results of handwritten recognition based on LSTM

最后我们采用Intel oneAPI AI组件实现端到端的模型优化，使用的优化工具为英特尔神经压缩器1.12，模型优化前后平均训练时间结果如下图所示。



测试了每批次训练数据分别为64和128情况下训练时间上的差异，结果如下图所示，结果显示经过优化后的网络可减少模型的训练时间。随着批次训练数据的增加，优化效果更加明显。

