摘　要

手写数字识别技术在医疗、邮政、金融、教育等领域具有广泛的应用前景，能够显著提高数据处理的效率和准确性。本论文围绕手写数字识别技术展开，系统地介绍了多种机器学习和深度学习模型在手写数字识别任务中的应用。

本文首先介绍了手写数字识别的背景和研究方法，然后详细阐述了逻辑回归、多层感知机、k 最近邻近、支持向量机、循环神经网络和卷积神经网络等模型的基本原理和在手写数字识别领域的应用特点。

通过在 MNIST 数据集上的实验对比，分析了不同模型的训练时间、测试时间和测试准确率。实验结果表明，卷积神经网络（CNN）在手写数字识别任务中表现最优，测试准确率达到了 99.34%。此外，本文还对 CNN 模型的不同参数进行了消融实验，分析了池化方法、卷积层数量、卷积核大小、Dropout 参数、全连接层大小等对模型性能的影响，为模型的优化提供了重要参考。

基于最优模型，本文开发了一个手写数字识别系统，部署在网页端，用户可以通过上传手写数字图片，实现手写数字的自动识别和结果展示。系统具备良好的用户体验，通过清晰的界面设计和简洁的操作流程，使用户能够便捷地使用手写数字识别功能。

最后，本文总结了手写数字识别技术的研究内容和意义，并对未来的发展方向进行了展望。未来可以进一步优化 CNN 模型的结构和参数，探索更高效的特征提取方法，提高模型的识别准确率和运行效率。同时，将手写数字识别技术应用于更多实际场景，如手写文本识别等，拓展其应用范围。

关键词：手写数字识别；深度学习；卷积神经网络（CNN）；Django 框架

@author Heineee

ABSTRACT

Handwritten digit recognition technology holds broad application prospects in healthcare, postal services, finance, education, and other fields, significantly enhancing the efficiency and accuracy of data processing. This thesis focuses on handwritten digit recognition technology and systematically introduces the application of various machine learning and deep learning models in this task.

The paper first outlines the background and research methodologies of handwritten digit recognition, followed by detailed explanations of the fundamental principles and application characteristics of multiple models, including Logistic Regression, Multilayer Perceptron (MLP), k-Nearest Neighbors (k-NN), Support Vector Machine (SVM), Recurrent Neural Network (RNN), and Convolutional Neural Network (CNN).

Through comparative experiments on the MNIST dataset, the training time, inference time, and test accuracy of different models were analyzed. Experimental results demonstrate that the Convolutional Neural Network (CNN) achieves optimal performance in handwritten digit recognition, with a test accuracy of 99.34%. Additionally, ablation experiments were conducted to investigate the impact of various CNN parameters—including pooling methods, number of convolutional layers, kernel sizes, Dropout parameters, and fully connected layer dimensions—on model performance, providing critical insights for model optimization.

Based on the optimal model, a web-based handwritten digit recognition system was developed. Users can upload handwritten digit images for automatic recognition and result visualization. The system delivers an excellent user experience through an intuitive interface design and streamlined operational workflow, enabling efficient utilization of the recognition functionality.

In conclusion, this paper summarizes the research contributions and significance of handwritten digit recognition technology while outlining future directions. Potential advancements include further optimization of CNN architectures and parameters, exploration of more efficient feature extraction methods, and improvements in recognition accuracy and computational efficiency. Furthermore, extending the technology to practical scenarios such as handwritten text recognition could broaden its applicability.

Key words: Handwritten digit recognition; Deep learning; Convolutional Neural Network (CNN); Django framework

@author Heineee

目 录

@author Heineee

第一章　引言

1.1 背景

手写数字识别技术作为人工智能图像识别领域的基石，在众多领域都发挥着重大作用[1]。它通过先进的算法和模型，能够将手写数字自动转换为电子数字，极大地提高了数据处理的效率和准确性。

在医疗领域[2]，手写数字识别可以快速将医生手写的病历和处方转化为电子文本，便于存储、检索和分析，减少了人工录入的工作量，同时也降低了因手写辨识不清而导致的错误。在邮政行业[2]，手写数字识别能够自动识别包裹上的手写邮编和地址信息，实现邮件的快速分拣和投递，提高了邮政服务的效率和准确性。在金融领域[2]，手写数字识别可以用于识别客户手写的支票金额和账号信息，减少了人工审核的工作量，提高了交易处理的速度和安全性。在教育领域[2]，手写数字识别可以辅助教师批改学生的数学作业，快速识别手写答案并进行评分，节省了教师的时间和精力，同时也为学生提供了及时的反馈。

总之，手写数字识别技术在各个领域的应用，不仅减少了人工录入和筛选的工作量，提高了数据处理的效率，还显著降低了人工的错误率，为现代社会的发展提供了有力的支持。

1.2 文献研究方法

随着机器学习和深度学习技术的迅猛发展，基于大数据的算法模型在手写数字识别领域展现出了极为优异的识别效果，极大地推动了该领域的进步[21][22]。

众多学者在这一研究热点上投入了大量的精力，先后提出了多种算法模型，包括基于逻辑回归(Logistic Regression)[16]，其以简洁的结构和高效的计算性能，在处理线性可分的数据集时能够取得较好的识别效果；多层感知机(Multi-Layer Perceptron, MLP)[3]，其通过构建多层神经元网络，能够学习到数据的非线性特征，从而提高识别准确率； k 最近邻近(k-Nearest Neighbors, KNN)[5]，其基于实例的学习方法，通过计算待识别样本与训练集中样本的距离，实现对手写数字的分类；支持向量机(Support Vector Machine, SVM)[8]，其通过寻找最优超平面，能够有效地处理高维数据，并在小样本情况下表现出良好的泛化能力；循环神经网络(Recurrent Neural Network, RNN)[18]，其利用对序列数据的处理能力，在手写数字识别中能够捕捉到数字的书写顺序信息和卷积神经网络(Convolutional Neural Network, CNN)[2]，其凭借其独特的卷积层和池化层结构，能够自动提取图像的局部特征，并通过多层卷积和池化操作逐步构建对图像的高层次理解，在手写数字

@author Heineee

识别任务中取得了极高的准确率。这些模型各具特点，在手写数字识别任务中均有良好的表现。

1.3 章节安排

本文将基于以上算法模型，进行模型的复现，同时验证 CNN 模型每个模块的性能提升和作用，并选用性能最好的模型构建手写数字系统的识别模型。具体而言，通过对各算法模型的实现与测试，深入探究其在手写数字识别任务中的表现，重点关注 CNN 模型中卷积层、池化层、全连接层等不同模块对整体性能的贡献，以便更好地理解其工作机制和优化方向。在完成模型验证后，将挑选出识别准确率最高、性能最为稳定的模型，作为手写数字系统的核心识别模型，以确保系统具备高效、准确的识别能力。

本文的第二部分介绍各模型算法理论，包括上述提及的基于逻辑回归(Logistic Regression)、多层感知机(Multi-Layer Perceptron, MLP)、k 最近邻近(k-Nearest Neighbors, KNN)、支持向量机(Support Vector Machine, SVM)、循环神经网络(Recurrent Neural Network, RNN)和卷积神经网络(Convolutional Neural Network, CNN)等模型。详细阐述每种模型的基本原理、算法流程及其在手写数字识别领域的应用特点，旨在为后续实验部分奠定坚实的理论基础，使读者能够深入理解各模型的本质和优势。

第三部分介绍基于以上算法实验对比以及基于 CNN 模型的消融实验。在实验对比环节，将利用相同的实验环境和数据集，对不同算法模型的性能进行全面评估，包括训练时间、测试时间以及测试准确率等关键指标，通过直观的数据对比，清晰地展示各模型在手写数字识别任务中的优劣。而在 CNN 模型的消融实验部分，将针对 CNN 模型的不同参数设置，如池化方法、卷积层数量、卷积核大小、Dropout 参数以及全连接层大小等，逐一进行测试和分析，探讨这些参数对模型性能的具体影响，从而为 CNN 模型的优化和选择提供有力依据。

第四部分介绍手写数字系统，主要基于第三部分训练好的 CNN 部署在网页端，实现与用户的交互界面和数据响应设计。具体介绍系统的设计理念、架构组成以及功能实现，包括用户如何上传手写数字图片、系统如何进行图片处理和识别，以及识别结果如何展示给用户等环节，重点突出系统的人性化设计和高效性，使用户能够便捷地使用手写数字识别功能。

第五部分总结。

本文整体研究思路框架如下图 1.1 所示：

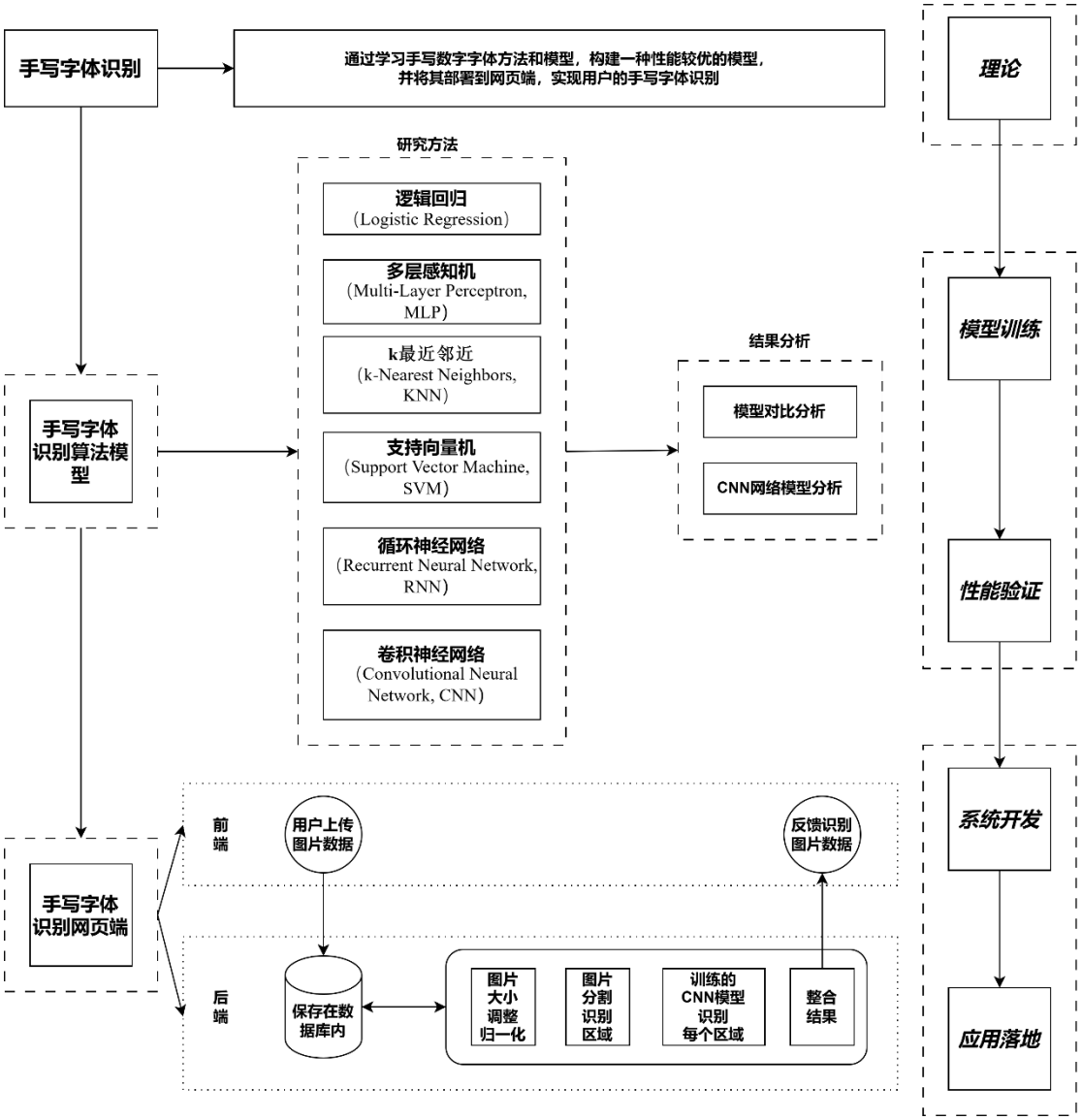

图 1.1 文章整体研究思路

第二章　理论方法

机器学习和深度学习的手写数字识别模型通常按照如下流程得出[3]，如下图 2.1 所示：

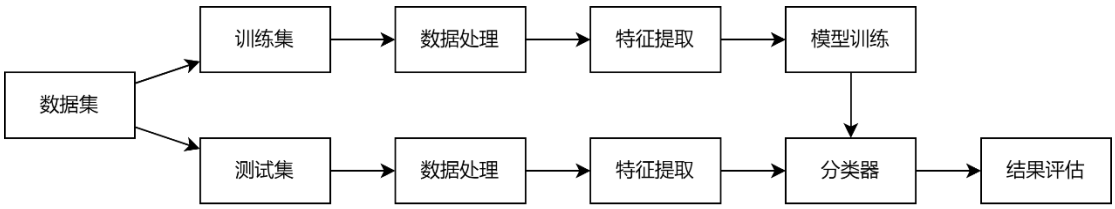

图 2.1 手写数字识别模型

其中，最为重要的一步就是分类器的构建，即模型的选择。

2.1 逻辑回归

逻辑回归(Logistic Regression)是一种广泛应用于统计学和机器学习领域的分类算法，尤其适用于分类问题，用于预测样本属于某个类别的概率[5]。

逻辑回归通过将线性回归的输出映射到一个概率值（0 到 1 之间），输出不同类别中可能性最高的一类，从而实现对样本的分类。其核心思想是使用逻辑函数，也称为 Sigmoid 函数，将线性回归的结果转换为概率，其表示为：

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{1}$$

其中，z 是线性回归的输出，即：

$$z = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n \tag{2}$$

其中，$(w_0, w_1, \ldots, w_n)$ 是模型不同特征的参数（权重），$(x_1, x_2, \ldots, x_n)$ 是输入特征，n为输入特征的维度。

逻辑回归的训练目标是找到一组最优的权重参数w，使得模型的预测结果与实际标签尽可能一致。逻辑回归的损失函数通常采用对数损失函数，也称为交叉熵损失函数：

$$J(w) = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log\left(\sigma(z_i)\right) + (1 - y_i)\log\left(1 - \sigma(z_i)\right)\right] \tag{3}$$

其中，N 是样本数量，y_i 是第 i 个样本的真实标签，$\sigma(z_i)$ 是模型预测的概率。

2.2 多层感知机

多层感知机(Multi-Layer Perceptron, MLP)是基于神经网络的分类器，其由三个不同的层组成：输入层、隐藏层和输出层。每个层都可以有一定数量的节点，也称为神经元，并且层中的每个节点都连接到下一层的所有其他节点，如下图 2.2 所示：

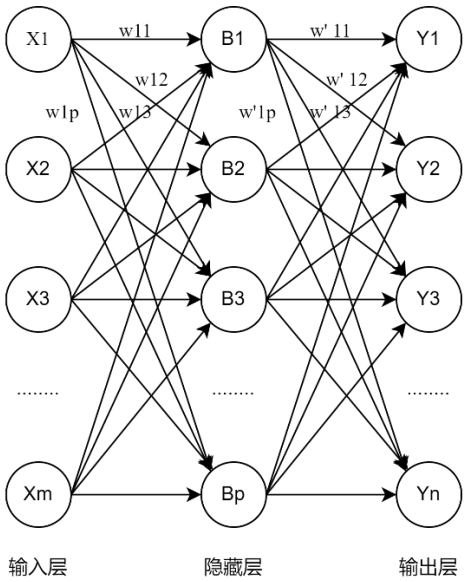

图 2.2 三层的 MLP 模型

输入层中的节点数取决于数据集的属性数量。输出层中的节点数取决于数据集类的数量。对于特定问题，隐藏层的节点数量很难确定通常通过具体实验选择。在多层感知器中，两个节点之间的连接由权重组成。在训练过程中，它使用反向传播算法的监督式学习技术以调节每个具体权重。

2.3 k 最近邻近

k 最近邻近算法(k-Nearest Neighbors, kNN)是一种用于分类和回归的非参数方法，它不需要学习大量的参数来构建模型，其核心思想是：给定一个样本，通过计算其与训练集中所有样本的距离（常用的距离包括欧氏距离、曼哈顿距离、闵可夫斯基距离、余弦相似度等），找到距离最近的 K 个样本，统计这 K 个样本中各类别的数量，将待分类样本归为数量最多的类别。

2.4 支持向量机

支持向量机(Support Vector Machine，SVM)是一种强大的监督学习算法，旨在通过最大化高维空间中类之间的边距来对数据点进行分类，其核心思想是找到

一个最优的超平面，将不同类别的样本分开，并且最大化两类样本之间的间隔。

当数据线性不可分时，SVM 引入松弛变量ξ_i，允许部分样本分类错误，同时通过惩罚参数C控制分类错误的程度。

优化目标：

$$\frac{1}{2}|w|^2 + C\sum_{i=1}^{n}\xi_i \tag{4}$$

约束条件：

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad 且 \quad \xi_i \geq 0 \tag{5}$$

其中，w 为权重向量，它定义了决策边界（超平面）的方向；C 为惩罚参数，用于控制模型对误分类的容忍度；ξ_i 为第 i 个数据点的松弛变量，表示第 i个数据点违反间隔约束的程度；y_i为第i 个数据点的标签；x_i为第 i 个数据点的特征向量，b为偏置项，它定义了决策边界的位置。

SVM 通过核函数将数据映射到高维空间，使其在高维空间中线性可分，常用的核函数[8]如下表 2.1 所示：

表 2.1　常用核函数

名称	表达式	说明
线性核	$\kappa(x_i, x_j) = x_i^{\top} x_j$	
多项式核	$\kappa(x_i, x_j) = (x_i^{\top} x_j)^d$	$d \geq 1$，多项式的次数
高斯核	$\kappa(x_i, x_j) = \exp\left(-\frac{\lvert x_i - x_j\rvert^2}{2\sigma^2}\right)$	$\sigma > 0$，高斯核的带宽
拉普拉斯核	$\kappa(x_i, x_j) = \exp\left(-\frac{\lvert x_i - x_j\rvert}{\sigma}\right)$	$\sigma > 0$
Sigmoid 核	$\kappa(x_i, x_j) = \tanh(\beta x_i^{\top} x_j + \theta)$	$\beta > 0, \theta < 0$，其中 $tanh$ 为双曲正切函数

2.5 循环神经网络

循环神经网络(Recurrent Neural Network, RNN)具有很强的挖掘数据时间信息和建模深度语义信息的能力，这使得它在语音识别、机器翻译等分析方面取得

不少最先进的性能。在时间步 t 处，RNN 将根据当前输入 x 输出一个输出隐藏向量 h。输出 h 将是时间步 $t+1$ 的输入，这使得顺序数据的建模成为一个循环过程，如下图 2.3 所示。

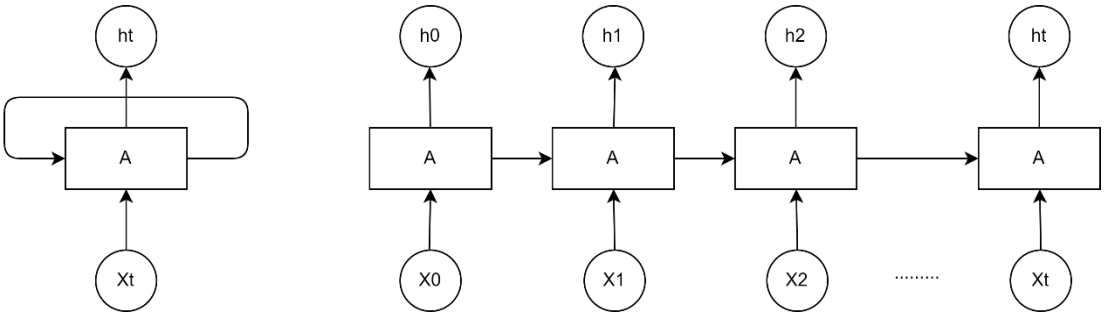

图 2.3 RNN 模型

2.6 卷积神经网络

卷积神经网络（Convolutional Neural Network, CNN）是神经网络的一种变体，它由神经元层组织，层之间只有少数神经元相互连接[24]。此外，CNN 是一种非线性模型，通常使用 RELU 或 Tanh 作为激活函数来学习更复杂的决策边界。一个 CNN 通常由以下 5 个模块组成：输入层、卷积层、池化层、全连接层和 softmax 层，如下图 2.4 所示：

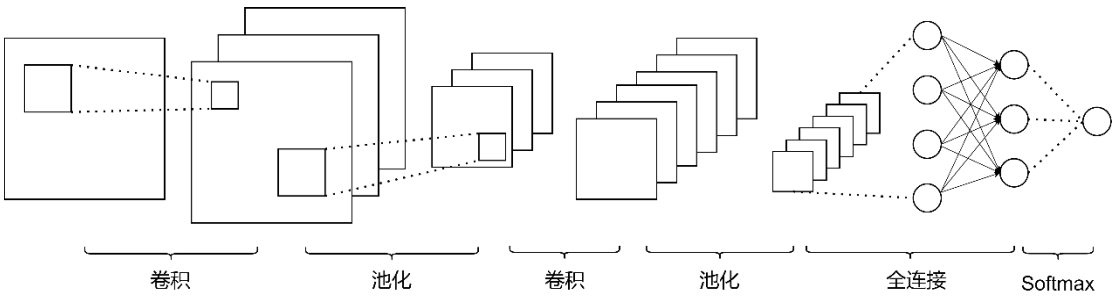

图 2.4 CNN 模型

2.6.1 卷积层

卷积层是 CNN 中最重要的层。它提取输入数据（如图像）的特征，通过卷积核（或滤波器）在输入数据上滑动，对局部区域进行加权求和，从而生成特征图。

卷积层输出的特征尺寸由卷积核的大小、步长和填充共同决定。假定卷积核的大小为 (h,w)，填充尺寸为 (p_1,p_2)，步长大小为 (s_1,s_2)，当输入的尺寸为 (c_1,c_2)，输出特征尺寸 (c_1',c_2') 可利用以下公式计算得出[9]：

$$c_1' = \frac{c_1 - h + 2p_1}{s_1} + 1 \tag{6}$$

$$c_2' = \frac{c_2 - h + 2p_2}{s_2} + 1 \tag{7}$$

在卷积层之后，通常会接一个激活函数，通过激活函数得到非线性输出，从而提高神经网络对非线性特征的拟合能力。常见的激活函数有 Sigmoid 激活（见公式 8）、Relu 激活（见公式 9）、Tanh 激活（见公式 10）等[11]。

$$Sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{8}$$

$$Relu(x) = \begin{cases} x & x > 0 \\ 0 & \text{其他} \end{cases} \tag{9}$$

$$Tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{10}$$

2.6.2 池化层

池化层可以降低特征图的空间尺寸，减少参数数量和计算量，同时保持特征的不变性[4]。常见的池化操作有最大池化(Max Pooling)和平均池化(Average Pooling)。最大池化通过选取局部区域的最大值来生成新的特征图，这有助于突出最重要的特征；平均池化则是计算局部区域的平均值, 即：

$$X = MaxPooling(X) \tag{11}$$

或者

$$X = AveragePooling(X) \tag{12}$$

2.6.3 全连接层

在卷积层和池化层提取特征后，全连接层用于将这些特征映射到输出空间，如分类任务中的类别概率，最终通过 Softmax 输出最终的分类结果, 即：

$$Output = Softmax\big(FullyConnectedLayer(X)\big) \tag{13}$$

2.7 本章总结

本章详细介绍了多种机器学习和深度学习模型的基本原理和在手写数字识别领域的应用特点。包括基于逻辑回归（Logistic Regression）、多层感知机（Multi-Layer Perceptron, MLP）、k 最近邻近（k-Nearest Neighbors, KNN）、支持向量机（Support Vector Machine, SVM）、循环神经网络（Recurrent Neural Network, RNN）和卷积神经网络（Convolutional Neural Network, CNN）等模型，这些理论和模型为后续实验奠定了坚实的理论基础。

第三章　模型选择与优化实验

3.1 模型任务

本章实验所用的模型是基于 PyTorch 框架自行开发的，旨在为手写数字识别模型的训练和测试提供一个稳定且高效的反馈，从而确保模型结果的准确性和可靠性。MNIST 手写数字识别是一项典型的分类任务[12][23][25]。实现的模型以 $28*28$ 图片的特征向量作为输入，并输出一个 0 到 9 之间的整数标签。整个 MNIST 数据集包括训练集图像和标签和测试集图像和标签，详情如下表 3.1 所示：

表 3.1 MNIST 数据集

数据集	文件名	数据量	说明
训练集图像	train-images-idx3-ubyte.gz	60000*28*28	6 万张手写数字图片
训练集标签	train-labels-idx1-ubyte.gz	60000	6 万张图片的标注
测试集图像	t10k-images-idx3-ubyte.gz	10000*28*28	1 万张手写数字图片
测试集标签	t10k-labels-idx1-ubyte.gz	10000	1 万张图片的标注

其中的训练集图片部分如下图 3.1 所示：

图 3.1 MNIST 数据集部分图片样例

令$X = x_1, x_2, \ldots, \mathrm{x_n}$($n$为$x$的数量)为一组输入的训练特征向量，$Y = y_1, y_2, \ldots, \mathrm{y_n}$($n$为$y$的数量)为训练样本标签，每一个$x$都是一组$D$维的向量。在 MNIST 数据集中，

n的大小为60000，D的大小为$28 * 28$。

3.2 实验环境

本代码是在 AMD Ryzen 7 4800H CPU 环境下运行，具体环境配置如下表 3.2 所示：

表 3.2 环境配置

Package	Version
numpy	1.23.5
scikit-learn	1.3.0
torch	2.0.1
torchvision	0.15.2
python	3.9.0

3.3 实验结果

利用 MNIST 数据集在逻辑回归(Logistic Regression)、多层感知机(Multi-Layer Perceptron, MLP)、k 最近邻近(k-Nearest Neighbors, KNN)、支持向量机(Support Vector Machine, SVM)、循环神经网络(Recurrent Neural Network, RNN)和卷积神经网络(Convolutional Neural Network, CNN)模型上运行 10 个 Epoch（如果模型需要 Epoch 训练）下得到的最优性能，如表 3.3 展示以下结果，具体模型参数如下所示：

（1）逻辑回归（Logistic Regression）：

Linear(in_features=784, out_features=10)，log_softmax（）

Adam 优化器，交叉熵损失函数，batch_size 设置为 512，Epoch 设置为 10。

其中，Linear(in_features=784, out_features=10) 是一个全连接层，将 784 维的输入特征向量映射到 10 维的输出特征向量。log_softmax() 是一个函数，将输入向量转换为对数概率分布，常用于多分类问题的输出层。

（2）多层感知机（Multi-Layer Perceptron, MLP）：

Linear(in_features=784, out_features=512) relu 激活函数

Linear(in_features=512, out_features=256) relu 激活函数

Linear(in_features=256, out_features=10) log_softmax（）

Adam 优化器，交叉熵损失函数，batch_size 设置为 512，Epoch 设置为 10。

@author Heineee

（3）k 最近邻近（k-Nearest Neighbors, KNN）：

K 迭代出最好效果为 3，其他参数选用 scikit-learn 中的默认值。

（4）支持向量机（Support Vector Machine, SVM）

最大迭代次数为 100，其他参数选用 scikit-learn 中的默认值。

（5）循环神经网络（Recurrent Neural Network, RNN）：

LSTM(input_size=28, hidden_size=64)

BatchNorm1d(64)

Dropout2d(p=0.25)

Linear(in_features=64, out_features=32) relu()

Dropout2d(p=0.5)

Linear(in_features=32, out_features=10) log_softmax（）

Adam 优化器，交叉熵损失函数，batch_size 设置为 512，Epoch 设置为 10。

其中，LSTM(input_size=28, hidden_size=64)：一个 LSTM 层，用于处理序列数据，每个时间步的输入特征维度为 28，隐藏状态维度为 64。BatchNorm1d(64)：对 LSTM 的隐藏状态进行批量归一化，归一化维度为 64。Dropout2d(p=0.25)：对二维数据进行随机丢弃，丢弃概率为 25%。

（6）卷积神经网络（Convolutional Neural Network, CNN）

Conv2d(1, 32, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2)),relu()

Conv2d(32, 32, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2)),relu()

MaxPool2d(kernel_size=2, stride=2)

Dropout(p=0.25)

Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1)),relu()

Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1)),relu()

MaxPool2d(kernel_size=2, stride=2)

Dropout(p=0.25)

Linear(in_features=3136, out_features=256),relu()

Linear(in_features=256, out_features=10), log_softmax（）

Adam 优化器，交叉熵损失函数，batch_size 设置为 512，Epoch 设置为 10。

其中，Conv2d(1, 32, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2)),relu()表示输入通道数为 1，输出通道数为 32，卷积核大小为 5×5，填充为 2，步长为 1。使用 ReLU 激活函数引入非线性。MaxPool2d(kernel_size=2, stride=2)表示使用 2×2 的池化窗口，步长为 2，将特征图的尺寸缩小一半。

表 3.3 各模型性能对比

模型	训练时间	测试时间	测试准确率
Logistic Regression	8.84s/Epoch	1.44s	92.45%
MLP	10.67s/Epoch	1.67s	98.23%
KNN	0.05s	5.46s	97.05%
SVM	46.64s	25.47s	93.02%
RNN	23.86s/Epoch	2.34s	97.65%
CNN	70.51s/Epoch	5.57s	99.34%

3.4 消融实验

为比较 CNN 的不同参数对模型参数的结果，现对 CNN 的不同参数测试其对结果的影响，各模型如下表 3.4 所示：

表 3.4 各模型及说明

模型	说明
CNN w/o Pool & CNN AveragePool	CNN 去除了池化和选择了其他的池化方法
CNN w/o layer	CNN 去除了第二次的卷积和池化操作
CNN channels	CNN 改变 channels 的大小
CNN linear	CNN 改变全连接层的大小
CNN kernel	CNN 改变卷积核的大小
CNN Dropout	CNN 改变 Dropout 的 p 大小
CNN	完整的实验 CNN 结构

由实验结果可知，在不影响性能的情况下，无论是平均池化还是最大池化都显著降低了模型的训练和推理时间；减少了第二次的卷积和池化操作，虽训练和推理时间大大减少，但性能稍微有点下降；dropout 的不同 p 值虽对训练时间和推理时间影响较小，但对模型的推理泛化存在一些影响；channels 的大小对训练时间和推理时间显著影响，也会对性能产生一些影响；全连接层的 linear 大小对训练时间和推理时间，性能影响较小；卷积核的大小对训练时间和推理时间显著影响，也会对性能产生一些影响。

以下图 3.2 展示了不同的池化方法对结果的影响，图 3.3 展示了减少了第二次的卷积和池化操作对结果的影响，图 3.4 展示了 dropout 的不同 p 值对结果的影响，图 3.5 展示了 channels 的不同对结果的影响，图 3.6 展示了全连接层的 linear 大小对结果的影响，图 3.7 展示了不同的卷积核大小对结果的影响。

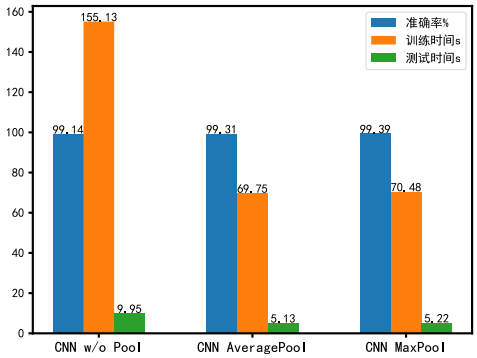

图 3.2 池化方法

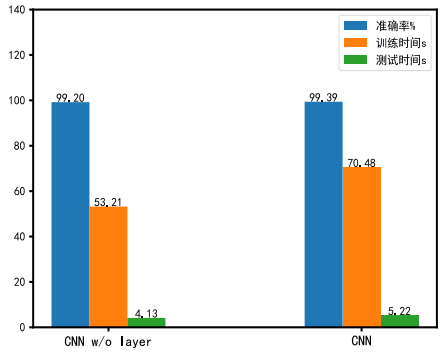

图 3.3 CNN 层数

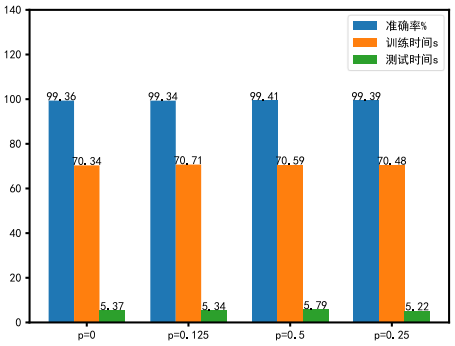

图 3.4 不同的 p 值

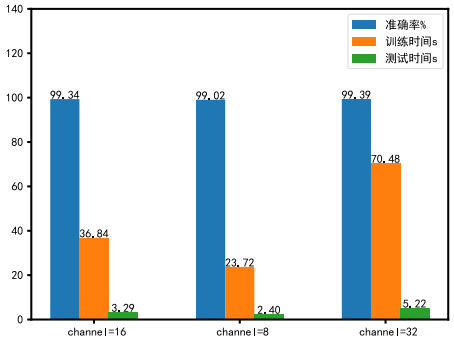

图 3.5 不同的 channels 值

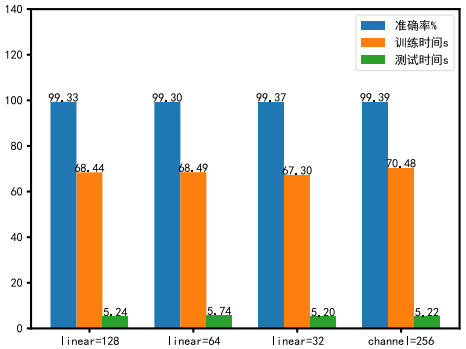

图 3.6 不同的 linear 值

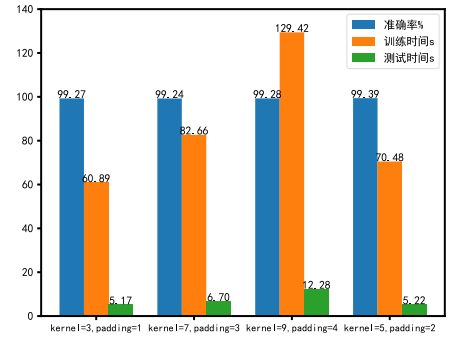

图 3.7 不同的卷积核大小

基于以上对算法性能和 CNN 参数的实验结果测试，最终选用以下模型作为系统的识别模型：

卷积神经网络（Convolutional Neural Network, CNN）：

Conv2d(1, 32, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2)),relu()

Conv2d(32, 32, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2)),relu()
MaxPool2d(kernel_size=2, stride=2)
Dropout(p=0.25)
Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1)),relu()
Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1)),relu()
MaxPool2d(kernel_size=2, stride=2)
Dropout(p=0.25)
Linear(in_features=3136, out_features=256),relu()
Linear(in_features=256, out_features=10), log_softmax（）
Adam 优化器，交叉熵损失函数，batch_size 设置为 512，Epoch 设置为 10。

3.5 本章总结

本章实验所用的模型是基于 PyTorch 框架自行开发的，旨在为手写数字识别模型的训练和测试提供一个稳定且高效的反馈，从而确保模型结果的准确性和可靠性。本章通过在 MNIST 数据集上进行实验，对比了多种机器学习和深度学习模型在手写数字识别任务中的性能。具体包括逻辑回归（Logistic Regression）、多层感知机（MLP）、k 最近邻近（kNN）、支持向量机（SVM）、循环神经网络（RNN）和卷积神经网络（CNN）。实验结果表明，卷积神经网络（CNN）在手写数字识别任务中表现最优，测试准确率达到了 99.34%。

此外，本章还对 CNN 模型的不同参数进行了消融实验，分析了池化方法、卷积层数量、卷积核大小、Dropout 参数、全连接层大小等对模型性能的影响。实验结果表明，不同的参数设置对模型的训练时间和推理时间有显著影响，同时也会对模型的性能产生一定的影响。例如，去除池化层或减少卷积层数量可以显著降低训练和推理时间，但可能会略微降低模型的性能；而改变卷积核大小和 Dropout 参数则对模型的泛化能力有一定的影响。

基于以上实验结果，最终选用了性能最优的 CNN 模型作为手写数字识别系统的识别模型。该模型在训练和测试中均表现出良好的性能，为后续手写数字系统的开发提供了坚实的基础。

第四章　手写数字系统

4.1Django 基本介绍及环境配置

本章基于第三章实验得出的最优 CNN 模型，进一步开发了一个手写数字识别系统。该系统不仅继承了实验模型的高效识别能力，还通过 Django[15]框架实现了与用户的交互，部署在网页端，使用户能够方便地使用手写数字识别功能，从而将理论研究转化为实际应用。

基于 Python 语言下的 Django 框架主要由模型、示图、模板、控制器四个部分构成[13][14]，之间的交互过程如下图 4.1 所示：

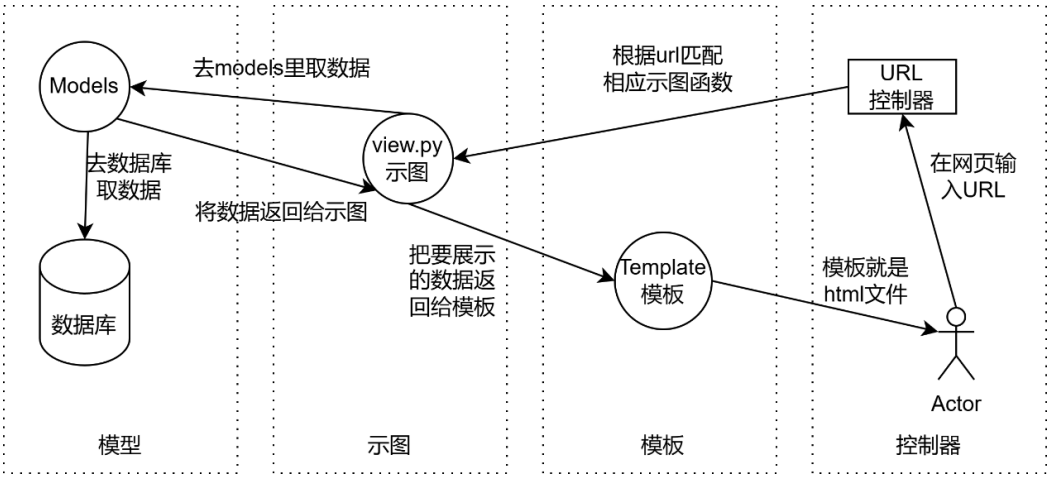

图 4.1 Django 框架

其中，模型（Model）负责处理与数据库的交互，定义数据的结构和行为。它决定了数据如何存储、检索和更新。

视图（View）负责处理业务逻辑，接收用户的请求并返回响应。视图是 Django 应用程序中处理用户请求的核心组件。例如接收请求、处理业务逻辑、返回响应等。

模板（Template）负责处理用户界面的呈现，生成 HTML 页面。模板是 Django 中用于生成动态网页内容的组件。

控制器（Controller）负责协调模型和视图之间的交互，处理用户的请求并返回响应。

本文所使用的开发环境如表 4.1 所示：

表 4.1 开发环境

Model	Version
Pycharm	2024.2.1
Django	4.2.17
python	3.9.0

4.2 网站功能和流程

该网站的主要功能是识别手写数字，并将识别结果展示给用户。用户可以上传包含手写数字的图片，系统会自动识别并标注每个数字。

1. 图片上传功能

用户可以通过点击“选择文件”按钮上传图片。须遵守图片格式、图片清晰度、图片内容、图片大小和版权与合法性五个方面：

（1）图片格式：仅接受 JPEG、PNG 格式图片，文件扩展名需与实际格式相符，例如 .jpg、.jpeg、.png。

（2）图片清晰度：图片应清晰可辨，分辨率不低于 300dpi，避免模糊、失真、有明显噪点或色彩偏差的情况，以确保文字能够被准确识别。

（3)图片内容：请确保图片中包含需要识别的手写字体内容，且文字部分完整、无遮挡，背景简洁，不会干扰字体识别。

（4）图片大小：单张图片文件大小不超过 2MB，若图片过大，请使用图片压缩工具进行适当压缩，以免影响上传和识别速度。

（5）版权与合法性：提交的图片应确保拥有合法版权，不侵犯他人知识产权或涉及任何违法违规内容。

2. 图片识别功能

上传的图片将通过后端的识别模型进行处理，识别出手写数字并返回结果。

3. 结果展示功能

识别结果将以标注的形式展示在图片上，用户可以直观地看到每个数字的识别结果。

在整体的识别过程中，数据处理和流向如下图 4.2 所示：

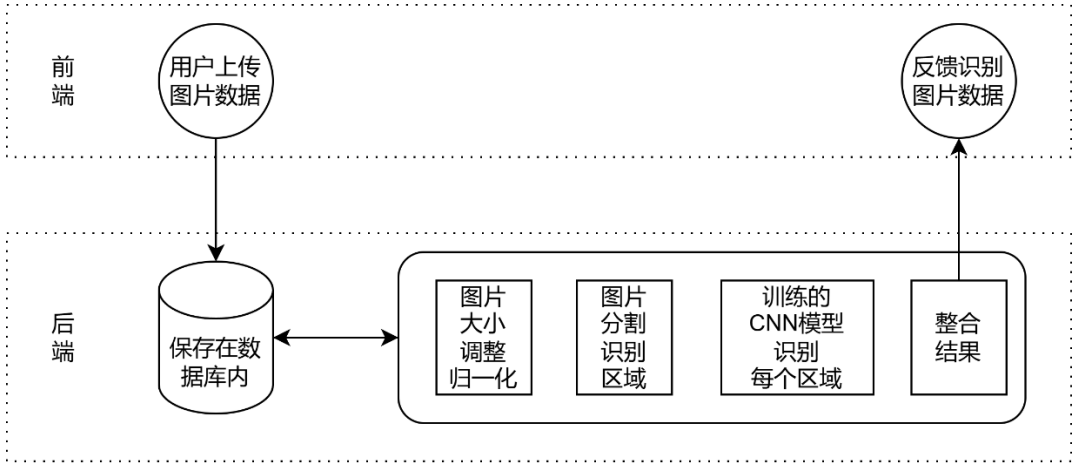

图 4.2 数据处理和流向图

4.3 网站功能演示

如下图 4.3 为网页的主页内容，涉及问题反馈和网页手写数字识别功能。

图 4.3 网页主页

按要求选择符合条件的图片并上传，如下图 4.4：

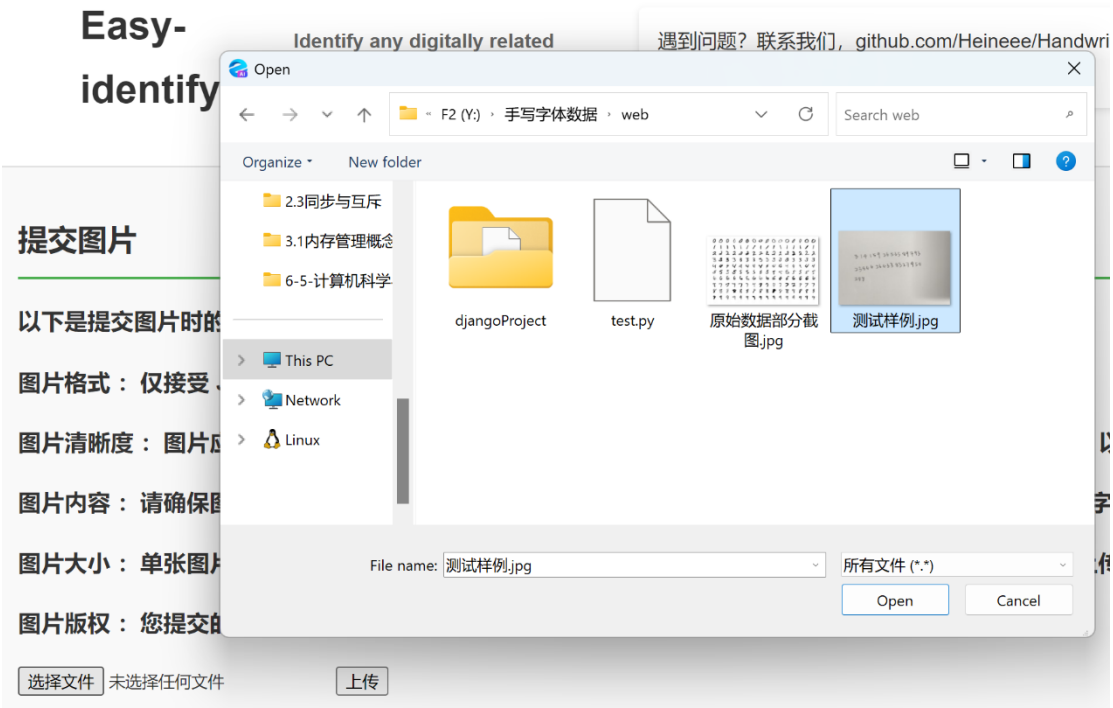

图 4.4 选择并上传需要识别的图片

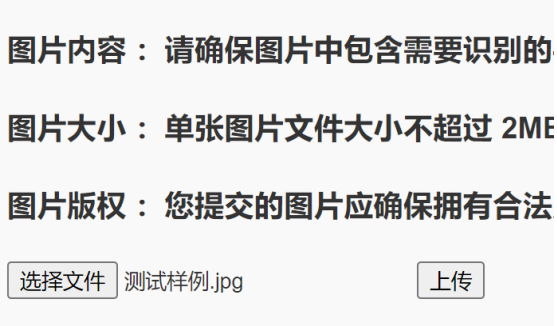

图 4.5 上传文件

测试样例. jpg 为测试图片，如下图 4.6 所示：

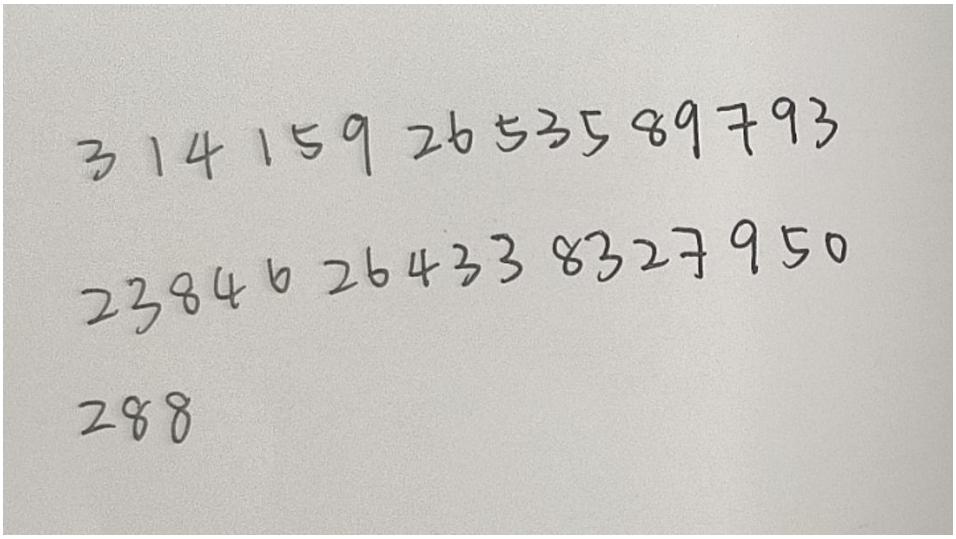

图 4.6 测试样例.jpg 图片

用户得到的反馈结果如下图 4.7（其中红色深色框内是手写的数字识别对象，绿色浅色数字是识别结果）所示：

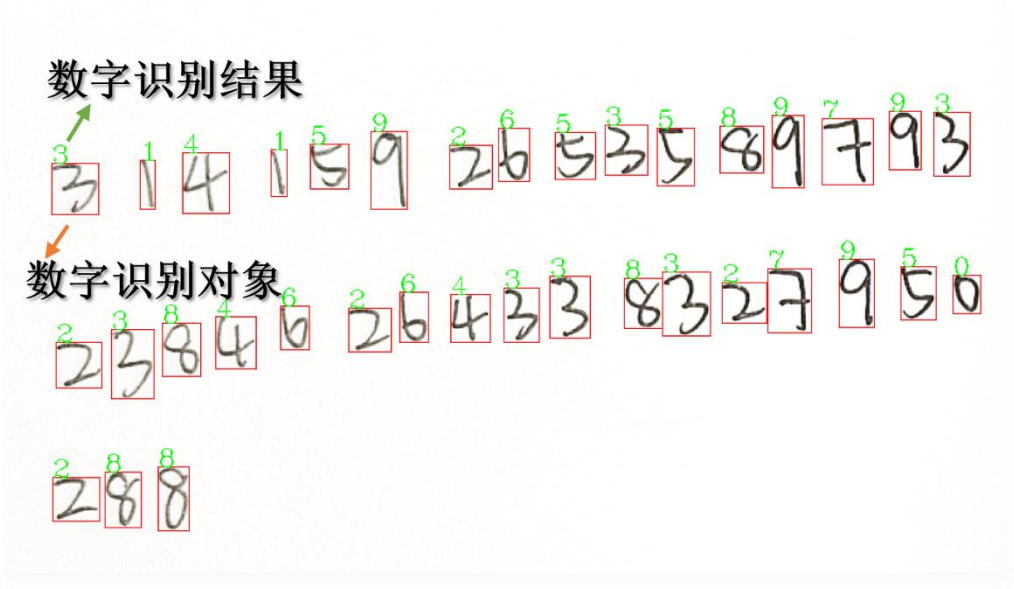

图 4.7 用户得到的反馈图

@author Heineee

由识别结果可以看出，上述结果无误。

4.4 系统创新性分析

本系统打通了从算法研究到实际应用的技术链路，系统设计的创新点包括：
1. 端到端集成：通过 Django 框架实现前后端高效交互，用户上传图片后，系统自动完成预处理、模型推理、结果标注全流程，无需人工干预。
2. 动态参数适配：针对不同分辨率的手写图片，系统内置图像归一化模块，确保输入数据与训练数据分布一致，提升识别稳定性。
3. 可视化增强：采用颜色区分标注（红色框标识识别区域，绿色数字展示结果），增强用户对识别过程的直观理解。
4. 轻量化部署优化：模型轻量化，在保持 99%以上准确率的同时，推理时间快速。

4.5 本章总结

本章基于第三章实验得出的最优 CNN 模型，进一步开发了一个手写数字识别系统。该系统不仅继承了实验模型的高效识别能力，还通过 Django 框架实现了与用户的交互，部署在网页端，使用户能够方便地使用手写数字识别功能，从而将理论研究转化为实际应用。具体功能包括：

（1）图片上传功能：用户可以上传符合要求的图片，系统支持 JPEG 和 PNG 格式，要求图片清晰、内容完整、大小不超过 2MB，并确保版权与合法性。

（2）图片识别功能：上传的图片通过后端的卷积神经网络（CNN）模型进行处理，识别出手写数字并返回结果。

（3）结果展示功能：识别结果以标注的形式展示在图片上，用户可以直观地看到每个数字的识别结果。

系统具备良好的用户体验，通过清晰的界面设计和简洁的操作流程，使用户能够便捷地使用手写数字识别功能。数据处理和流向图展示了从图片上传到结果展示的整个过程，确保了系统的高效性和准确性。

第五章　总结

本文围绕手写数字识别技术展开，系统地介绍了多种机器学习和深度学习模型在手写数字识别任务中的应用，并通过实验对比分析了各模型的性能。同时，基于最优模型构建了一个手写数字识别系统，实现了网页端的手写数字识别功能。

5.1 具体研究内容

1. 理论方法：本文详细介绍了多种手写数字识别模型，包括逻辑回归、多层感知机、k 最近邻近、支持向量机、循环神经网络和卷积神经网络。特别地，对卷积神经网络的各个模块（卷积层、池化层、全连接层）进行了深入探讨，分析了其在特征提取和分类中的作用。
2. 实验对比：通过在 MNIST 数据集上进行实验，对比了不同模型的训练时间、测试时间和测试准确率。实验结果表明，卷积神经网络（CNN）在手写数字识别任务中表现最优，测试准确率达到了 99.34%。
3. 消融实验：对 CNN 模型的不同参数进行了消融实验，分析了池化方法、卷积层数量、卷积核大小、Dropout 参数、全连接层大小等对模型性能的影响。实验结果为模型的优化提供了重要参考。
4. 手写数字系统：基于 Django 框架开发了一个手写数字识别系统，用户可以通过网页上传手写数字图片，系统自动识别并展示结果。系统具备图片上传、图片识别和结果展示等功能，用户体验良好。

5.2 研究意义

1. 提高效率：手写数字识别技术在医疗、邮政、金融、教育等领域具有广泛的应用前景，能够显著减少人工录入和筛选的工作量，提高数据处理效率。
2. 降低错误率：通过机器学习和深度学习模型，手写数字识别的准确率得到了显著提升，从而降低了人工录入的错误率。
3. 技术推动：本文的研究不仅验证了现有模型的有效性，还通过消融实验为模型的进一步优化提供了依据，推动了手写数字识别技术的发展。

5.3 未来展望

由于对本课题接触的时间有限，以及本人对算法模型设计和系统设计方面知识有所欠缺，本项目仍然存在许多可以优化的地方。以下是对于项目的几点展望：

1. 模型优化：未来可以进一步优化 CNN 模型的结构和参数，探索更高效的特征提取方法，提高模型的识别准确率和运行效率。
2. 多场景应用：将手写数字识别技术应用于更多实际场景，如手写文本识别等，拓展其应用范围。
3. 用户交互：改进手写数字识别系统的用户界面和交互设计，提升用户体验，使其更加友好和便捷。
4. 多模态融合：结合其他模态数据（如语音等），探索多模态融合的手写数字识别方法，进一步提高识别的准确性和鲁棒性。
5. 系统性能优化：在系统性能方面，虽然本项目已经实现了基本的功能，但在处理大规模数据时仍存在一定的性能瓶颈。未来可以对系统的架构和算法进行优化，例如采用更高效的数据库存储方案、优化数据处理流程等，以提高系统的整体性能和响应速度。
6. 安全性增强：未来可以加强对系统的安全性设计，例如增加用户认证、数据加密等功能，以保护用户的隐私和数据安全。

参考文献

[1] 余国庆, 杨燕婷, 宗兆星, 等. 基于卷积神经网络的手写数字识别技术研究[J]. 安徽电子信息职业技术学院学报, 2024,23(03):1-5.

[2] 刘辰雨. 基于卷积神经网络的手写数字识别研究与设计[D]. 成都理工大学, 2018.

[3] 张华美, 张皎洁. 基于人工智能的脱机手写数字识别研究综述[J]. 南京邮电大学学报(自然科学版), 2021,41(05):83-91.DOI:10.14132/j.cnki.1673-5439.2021.05.012.

[4] 钟乐海, 胡伟. 手写体数字识别系统中一种新的特征提取方法[J]. 四川大学学报(自然科学版), 2007,(05):1000-1004.

[5] 卢利琼, 吴东. 一种利用 KNN 实现手写数字识别的方法[J]. 现代信息科技, 2021,5(04):97-99.DOI:10.19850/j.cnki.2096-4706.2021.04.024.

[6] 汪愿. 基于神经网络的手写数字图像识别研究设计[J]. 电工材料,2021,(06):46-48.DOI:10.16786/j.cnki.1671-8887.eem.2021.06.012.

[7] 郑继燕. 基于 CNN 的手写数字识别与试卷管理系统设计[D]. 北京邮电大学, 2020.DOI:10.26969/d.cnki.gbydu.2020.001672.

[8] 李雅琴. SVM 在手写数字识别中的应用研究[D]. 华中师范大学, 2007.

[9] 蒙庚祥, 方景龙. 基于支持向量机的手写体数字识别系统设计[J]. 计算机工程与设计, 2005,(06):1592-1594+1598.DOI:10.16208/j.issn1000-7024.2005.06.062.

[10] 李朝. 基于时空卷积神经网络的流量预测技术研究[D]. 北京邮电大学, 2023.DOI:10.26969/d.cnki.gbydu.2023.002669.

[11] 张贯航. 基于 MNIST 数据集的激活函数比较研究[J]. 软件,2023,44(09):165-168.

[12] 唐鉴波, 李维军, 赵波, 等. 基于卷积神经网络的手写数字识别方法研究[J]. 电子设计工程, 2022,30(21):189-193.DOI:10.14022/j.issn1674-6236.2022.21.040.

[13] 路明玉. 手写数字识别的系统设计[J]. 科技资讯, 2019,17(19):31+33.DOI:10.16661/j.cnki.1672-3791.2019.19.031.

[14] 王亚威. 手写体数字识别系统的设计与实现[D]. 河北科技大学, 2015.

[15] 陈红梅, 李晟, 李玉晓. 基于 STM32 的手写数字识别平台的设计与实现[J].现代信息科技, 2023,7(21):63-66+70.DOI:10.19850/j.cnki.2096-4706.2023.21.015.

[16] 陈玲. 基于 Django 的名著人物展示系统的设计与实现[J]. 中国信息界,2024,(07):188-190.

[17] Chen S, Ahmmed S, Lal K, et al. Django web development framework: Powering the modern web[J]. American Journal of Trade and Policy, 2020, 7(3): 99-106.

[18] Moolchandani J, Kumar R, Singh K. Advancements in Handwritten English Character Recognition: A Comprehensive Analysis Using Pattern Recognition and Deep Learning[C]//Proceedings of the Second Congress on Control, Robotics, and Mechatronics: CRM 2024, Volume 1. Springer Nature, 319.

[19] Shamim S M, Miah M B A, Sarker A, et al. Handwritten digit recognition using machine learning algorithms[J]. Indonesian Journal of Science and Technology, 2018, 3(1): 29-39.

[20] Shrivastava A, Jaggi I, Gupta S, et al. Handwritten digit recognition using machine learning: A review[C]//2019 2nd International Conference on Power Energy, Environment and Intelligent Control (PEEIC). IEEE, 2019: 322-326.

[21] Muthureka K, Srinivasulu Reddy U, Janet B. Noise filtering approach to improve handwritten digit recognition using customized CNN for Cerebral Palsy individuals[J]. The European Physical Journal Special Topics, 2025: 1-19.

[22] Jabde M, Patil C H, Vibhute A D, et al. A systematic review of multilingual numeral recognition systems[J]. Artificial Intelligence Review, 2025, 58(4): 106.

[23] Rahman A B M A, Hasan M B, Ahmed S, et al. Two decades of bengali handwritten digit recognition: A survey[J]. IEEE Access, 2022, 10: 92597-92632.

[24] Haghighi F, Omranpour H. Stacking ensemble model of deep learning and its application to Persian/Arabic handwritten digits recognition[J]. Knowledge-Based Systems, 2021, 220: 106940.

[25] Beohar D, Rasool A. Handwritten digit recognition of MNIST dataset using deep learning state-of-the-art artificial neural network (ANN) and convolutional neural network (CNN)[C]//2021 International conference on emerging smart computing and informatics (ESCI). IEEE, 2021: 542-548.