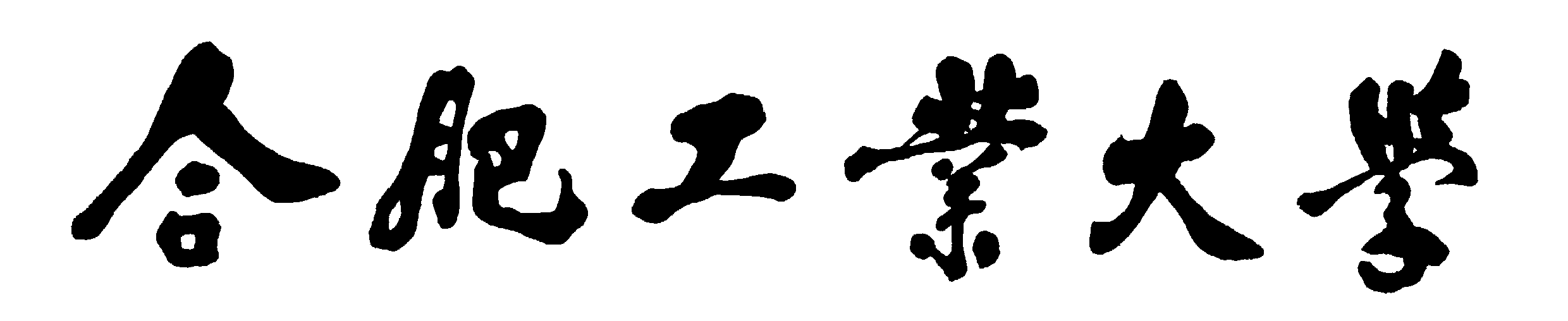
****

（计算机与信息学院）

机器视觉实验报告

|  |  |  |
| --- | --- | --- |
| 专业班级 | | 智能科学与技术21-1班 |
| 学生姓名学号 | | 郭宇童2021214582 |
| 任 课 教 师 | | 洪日昌 |
| 实验名称 | | 实验3学号识别 |
| 2023-2024学年第 1 学期 | |

# 引言

本实验旨在使用MNIST手写数字数据集进行数字识别任务，展示了机器学习和计算机视觉在解决实际问题中的应用。MNIST数据集包含60,000个训练样本和10,000个测试样本，每个样本是一个28x28像素的灰度手写数字图像。

实验的主要目的是开发和评估能够准确识别手写数字的机器学习模型。通过这个过程，实验旨在深入了解图像处理、特征提取、模型训练和优化等关键概念。

实验过程包括以下几个阶段：

* 数据预处理：加载MNIST数据集，将图像转换为模型可处理的格式，并进行归一化处理。
* 模型构建：设计用于识别手写数字的机器学习模型。实验中采用了**多种模型**，包括全连接神经网络和卷积神经网络（CNN），以比较它们的性能。
* 训练与验证：使用训练数据集训练模型，并在测试集上验证模型的准确性。
* 结果分析：评估模型在手写数字识别任务上的性能，通过准确率、损失函数等指标进行分析。

# 1 minist数据集介绍

MNIST（Modified National Institute of Standards and Technology）数据集是机器学习领域最著名的数据集之一，特别是在图像处理和手写数字识别的应用上。这个数据集常被用作计算机视觉和机器学习算法的基准测试。

## 1.1 基本信息

* 内容：MNIST数据集包含了大量的手写数字图像，分为训练集和测试集。它包含了从0到9的10个数字的手写样本。
* 规模：数据集包含60,000个训练样本和10,000个测试样本。
* 图像大小：每个图像都是28x28像素的灰度图。
* 标签：每个图像都有一个与之对应的标签，表示图像中手写数字的真实值。

## 1.2 数据集应用

* 基准测试：MNIST数据集广泛用于评估和比较不同的图像识别、机器学习和深度学习模型的性能。
* 教育用途：由于其相对简单且处理方便的特性，MNIST数据集常被用作教学和入门级机器学习项目的数据集。

## 1.3 数据集特点

* 格式统一：所有图像都被格式化为相同的大小，且已经过灰度处理，便于直接使用在各种算法中。
* 多样性：尽管图像是单一的手写数字，但包含了多种不同的书写风格，这对于训练鲁棒性较强的模型十分有用。
* 广泛支持：由于其在机器学习领域的知名度，多数机器学习框架都直接支持从官方接口下载和加载MNIST数据集。

# 2实验实现模型简介与对比

## 2.1 第一个模型：全连接层网络

### 2.1.1 网络架构

* 该模型首先定义了一个Sequential模型，表示模型的层会按顺序堆叠。
* 第一个Dense层有121个神经元，激活函数为ReLU。它接收输入数据的维度为28\*28（即展平的MNIST图像）。
* 第二个Dense层作为输出层，有10个神经元，对应于10类数字的分类，使用softmax激活函数。

### 2.1.2 编译和训练：

* 模型使用categorical\_crossentropy作为损失函数，sgd（随机梯度下降）作为优化器。
* 模型在训练数据上训练30个epochs，每个batch包含64个样本。
* 使用验证数据（测试集）进行模型验证。

## 2.2 第二个模型：卷积神经网络（CNN）

### 2.2.1 网络架构：

* 这个模型也是一个Sequential模型。
* 首先是两个卷积层（Conv2D），分别有32个和64个卷积核，激活函数为ReLU。
* 每个卷积层后跟一个最大池化层（MaxPooling2D）。
* 然后是一个Flatten层，将卷积层的输出展平，以便传递给全连接层。
* 接着是一个有128个神经元的全连接层（Dense layer），激活函数为ReLU。
* Dropout层用于减少过拟合。
* 最后是一个有10个神经元的全连接层，激活函数为softmax。

### 2.2.2 编译和训练：

* 模型使用categorical\_crossentropy作为损失函数，Adadelta作为优化器。
* 模型在训练数据上训练10个epochs，每个batch包含128个样本。
* 使用验证数据（测试集）进行模型验证。

## 2.3 两个网络的比较

* 网络架构：第一个模型是一个简单的全连接网络，而第二个模型是一个卷积神经网络。CNN通过卷积层能够更好地捕获图像中的空间层次结构和模式，通常在图像识别任务中表现更优。
* 参数数量：CNN由于其权重共享机制，通常有较少的参数，更高效。
* 性能：一般来说，在图像识别任务中，CNN比全连接网络有更好的性能，尤其是在处理较大或较复杂的图像数据集时。
* 适用性：全连接网络适用于较为简单的、非空间层次性的问题，而CNN更适合处理具有明显空间结构的图像数据。

# 3 模型其他部分实现

## 3.1 数据预处理（以卷积为例）

### 3.1.1 读取标签（read\_labels 函数）：

该函数从二进制文件中读取图像的标签（即手写数字的实际值）。

使用open函数以二进制读取模式打开标签文件。

file\_labels.seek(8) 跳过文件头的8个字节，因为这部分通常包含与数据集格式相关的元数据。

然后，读取指定数量（items）的标签数据，并将它们存储在一个NumPy数组中。

### 3.1.2 读取图像数据（read\_images 函数）：

类似地，这个函数从二进制文件中读取图像数据。

file\_image.seek(16) 跳过前16个字节的文件头。

读取图像数据并将每个像素的值除以255。这个步骤称为归一化，它将像素值从范围[0, 255]转换到[0, 1]，有助于模型训练的稳定性和效率。

最后，使用reshape方法将一维数组转换为4D张量，以适配CNN模型的输入。每个图像被重塑为(28, 28, 1)，表示28x28像素的灰度图像。

### 3.1.3 加载训练和测试数据：

使用上述函数加载训练和测试数据集。对于训练数据集（X\_train, y\_train），读取60000个样本；对于测试数据集（X\_test, y\_test），读取10000个样本。

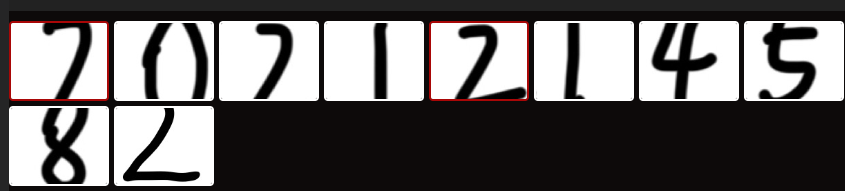
### 3.1.4 One-Hot 编码：

keras.utils.to\_categorical 函数用于将类别向量（整数）转换为二进制类矩阵，这在处理分类问题时非常有用。

对于每一个样本的标签，原始的单个整数被转换为一个长度为10的向量，其中对应于类别的索引位置为1，其余为0。例如，数字3被转换为[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]。

## **3.2 图像分割**

由于任务为实现手写数字串的识别，因此要对图像进行分割。

  
图 分割后的图像

* 打开图像：

使用 Image.open 来加载指定路径的图像。

* 获取图像尺寸：

image.size 用于获取图像的宽度和高度。

* 计算单个数字的宽度：

假设图像中包含10个数字，且每个数字的宽度相同，所以将图像的总宽度除以10得到每个数字的宽度。

* 创建输出文件夹：

使用os.makedirs创建输出文件夹。exist\_ok=True参数意味着如果文件夹已存在，则不会引发错误。

* 分割并保存数字：

循环遍历10个数字的位置，使用图像的crop方法进行分割。分割区域由每个数字的左右边界（left, right）和图像的高度定义。

将分割后的每个数字图像保存到指定的输出文件夹中，文件名为数字的序号（0.png, 1.png, ..., 9.png）。

* 脚本主体：

在脚本的主执行部分，定义了输入图像的路径和输出文件夹的路径。

调用split\_and\_save\_digits函数来执行分割和保存操作。

执行完成后，打印一条消息表示操作完成。

# 4 模型训练过程与识别结果

手写数字原图如下

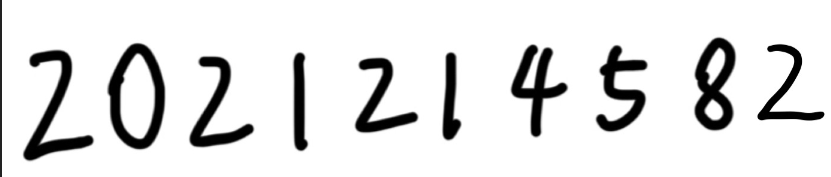


图 学号手写原图

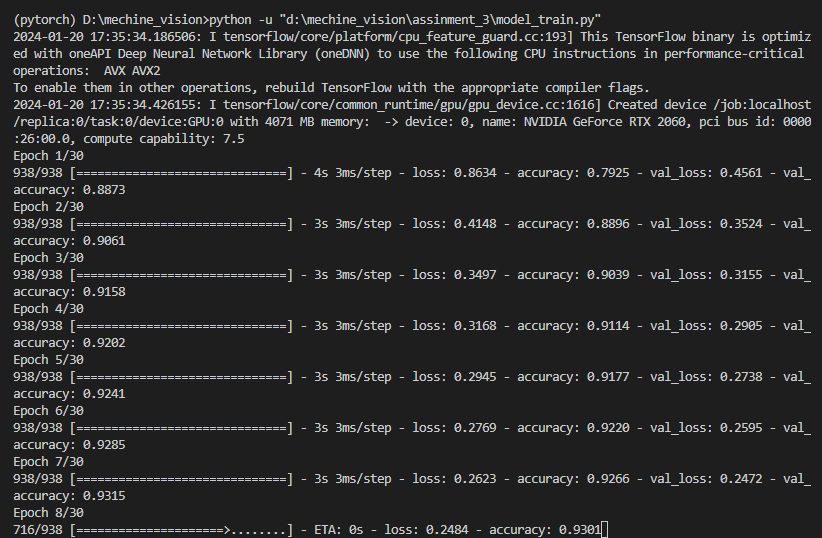


图 训练过程（RTX2060）

模型识别结果如下

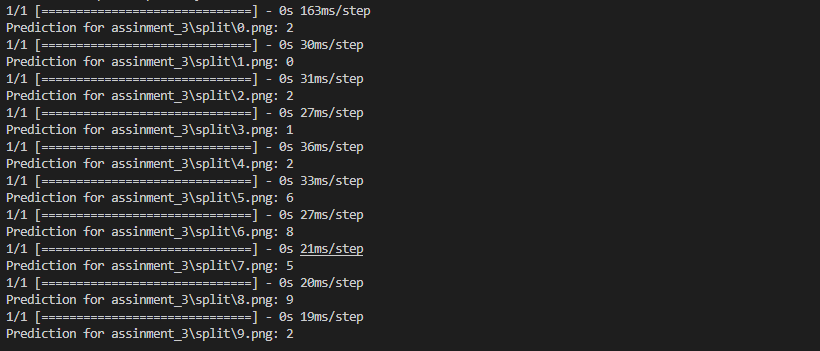


图 模型识别结果

根据识别结果有几处出现了错误，估计是与中外数字书写方式差异有关，其中有一个数字为1，但是写的比较像6，所以识别成了6，见下图：

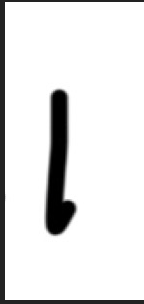


图 手写数字