**A + B赛题技术报告**

4\_杨致远\_20231301004

A+B赛题提供了10000张测试图片（inputs）和2023张带注释的图片（samples）。注意到samples中的图片和测试图片相似度很高，所以我选择针对性利用samples中的图片，对其进行处理后训练模型，最后进行测试。我的具体实现途径分为以下四个步骤。

**一、利用opencv提取单个数字（****cvpreproctest.py）**

samples和inputs中的图片都是96\*48像素，包含两个手写数字，因此我需要利用opencv相关技术将两个数字单独分别提取，将数字单个化后再进行识别。我在cvpreproctest.py中创建preproc1类，首先将图片进行灰度化、二值化处理，按位取反（将数字部分变成白色），随后对数字进行形态学上的闭操作（close），填充图像的凹角，对其进行外部轮廓查找（findcontours），同时对两个数字进行从左到右的排序，避免两个数字位置错乱。最后preproc1类的返回值包含一个列表，列表内有2个从图片上拆分出来的数字。

**二、对samples中的图片进行批处理（batch.py）**

batch.py用于对samples中2023张图片进行批量化处理。在2023次循环中，每一次都先将图片用步骤一的技术一分为二，同时为了匹配神经网络训练时对图片的要求，将含有单个数字的图片调成28\*28像素（resize），保存到指定文件夹。随后将explanations中对应的数值信息（每个文本文件中的两个数字）也拆分成两个文本文件保存。至此，就得到了samples中的4046个单个数字图片和与其一一对应的数值。

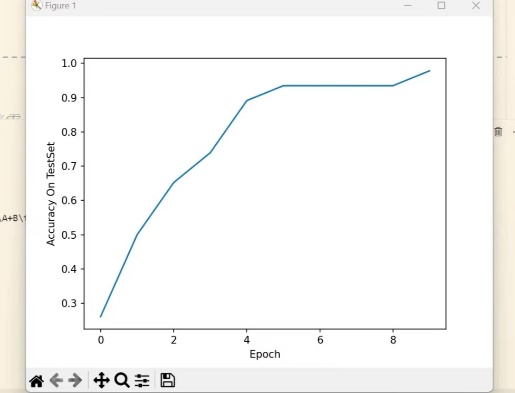
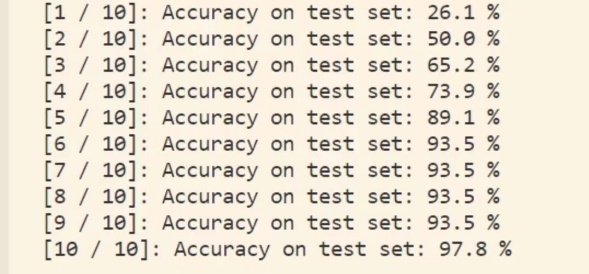
**三、训练模型（train.py）**

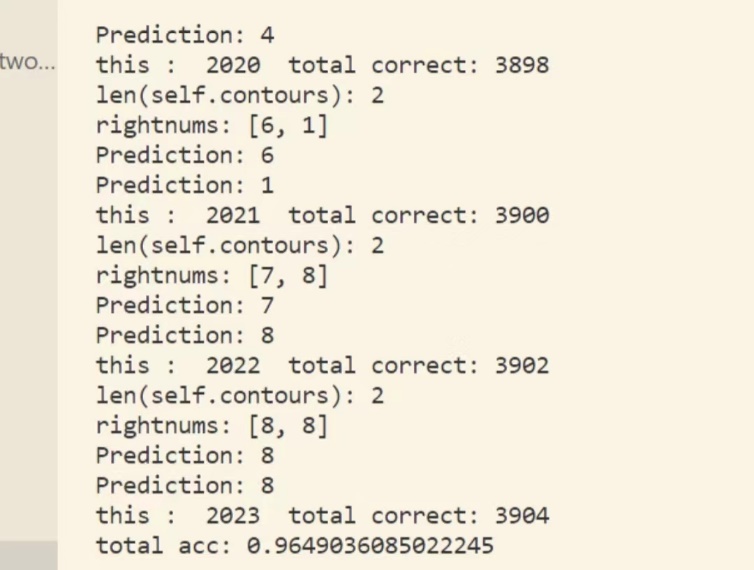
将上一步获得的4046个单个数字图片和与其一一对应的数值文本拆分成4000组图片+文本，分别保存在splited\_img文件夹和splited\_label文件夹中，作为训练集；剩余46组图片+文本分别保存在splited\_img\_test文件夹和splited\_label\_test文件夹中，作为测试集。

在train.py中，我先定义两个数据集（训练集和测试集），每调用一次数据集都返回一张图片和与之对应的标签，此时的图像已经转换为了张量（ToTensor）。

分别初始化训练集、训练加载器、测试集、测试加载器，随后定义一个CNN卷积神经网络，通过卷积、池化、激活等操作后变为全连接网络，最后输出维度为10（对应数字0~9）。

定义损失计算器和参数优化计算器，在训练过程中把单独的一轮封装在函数类中，每一轮都变量初始化，之后读取训练集，先前向传播一次，计算损失值；再反向传播，优化器进行一步。随后累加计算损失值，计算训练过程中每一轮的准确率。每训练一轮都把模型（model\_AB.pth）和优化器参数（optimizer\_AB.pth）保存到本地。

同时，每训练一轮，都使用这一轮中的模型对测试集进行测试，并输出正确率：

从训练结果可知，训练到第10轮时对测试集的测试正确率已经达到了97.8%，说明模型的质量已经达到了要求，可以进一步用这个模型推理inputs中的10000张图片。（我还用训练好的模型测试了samples中的4046张图片，结果显示3904张图片识别正确，正确率为96.49%）

**四、最终识别（pred\_recog.py）**

获得网络结构，加载模型，把模型转化为评估模型，每次循环都处理inputs中的一张图片，将两个数字的识别结果相加作为答案写入对应的文本文件，保存到outputs文件夹。