交通信号灯分类

姓名：尹伯豪 学号：2112215089

文章采用了CNN卷积神经网络和SVM两种算法来对所给数据集进行训练并分类，数据集的读取采用了pytorch的ImageFolder函数，读取的同时对图像数据进行了归一化，标准化处理，并通过DataLoader函数，将训练集和测试集分批加载进内存，达到了快速读取的目的。

**CNN卷积神经网络图像多分类：**

* 基本原理：通过使用不同的卷积核扫描图像来识别边缘，识别纹理，提取出抽象的特征，最终实现图像识别，并通过损失函数来优化模型参数，从而实现对输入数据的分类任务。

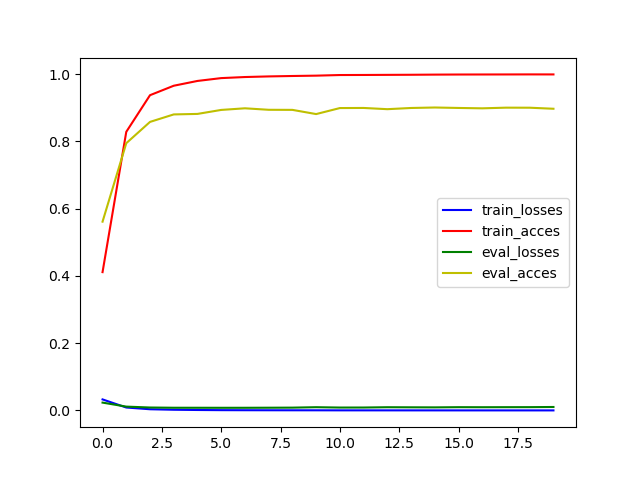
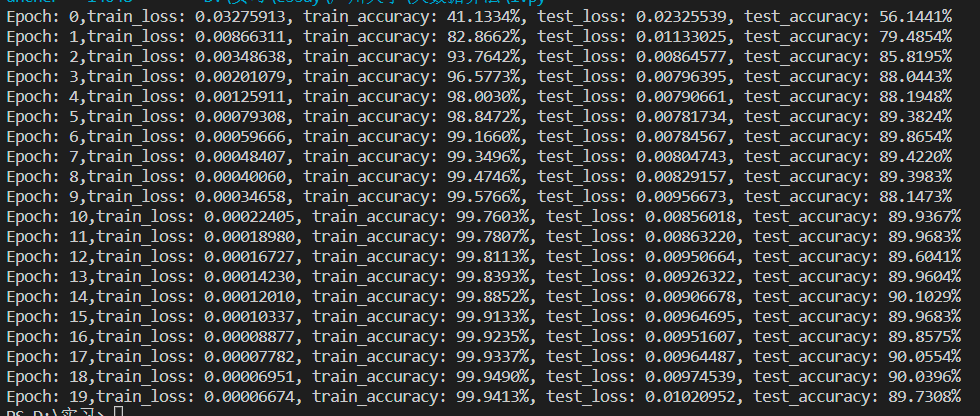
文章采用了两层卷积，其中每层包含了卷积，Relu激活函数和最大值池化：

nn.Conv2d(in\_channels=3, out\_channels=32, kernel\_size=3, padding=1),

nn.ReLU(),

nn.MaxPool2d(kernel\_size=2, stride=2),

第一版测试中，神经网络内包含了一层输入，两层卷积和一层输出，结果导致训练集的拟合率达到99.9%，产生了过拟合，从而使得测试集预测结果产生波动，并且准确率达不到90%。



为了解决训练集过拟合现象，采用了正则化技术，首先是通过L1和L2正则化向损失函数中添加一个与网络权重相关的正则项，使得网络权重变小，从而防止过拟合。

optimizer = torch.optim.SGD(model.parameters(), lr=0.03, weight\_decay=0.01)

接着在神经网络内的前向传播中加入Dropout函数，在每个训练迭代中随机的将一些神经元设置为0，从而使得拟合程度不会过度依赖某些特定的神经元。

nn.Dropout(0.5)

除此之外，还采用了动态学习率的方式，通过每五轮训练，学习率降低10%来抑制训练后期的过拟合情况。

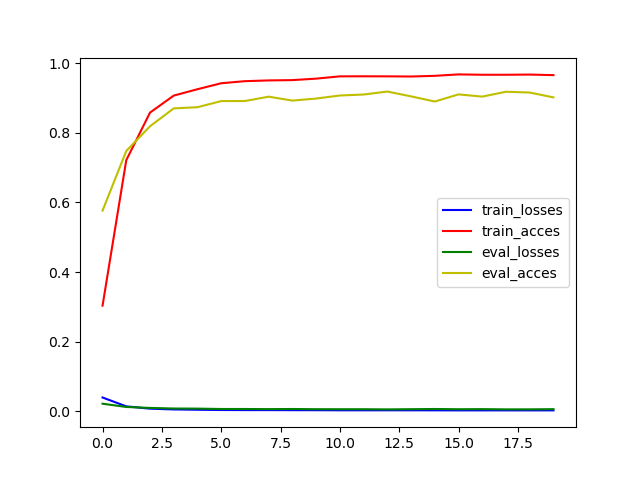
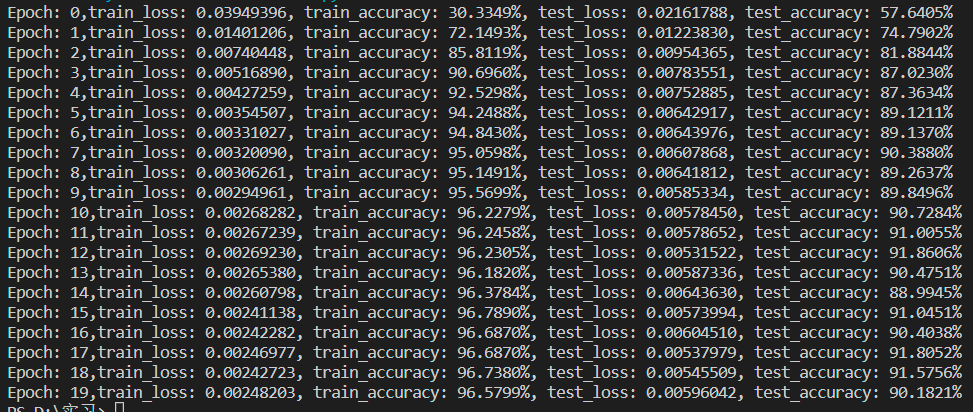
if epoch % 5 == 0: optimizer.param\_groups[0]['lr'] \*= 0.9

第二版测试结果显示，训练集的拟合率从99.9%降到了96%，并且测试集的准确率也在10轮训练之后稳定在了90%以上，最高可达91.8%，初步达到了训练分类任务。

最终版参数的选取：

batch\_size = 64, learning\_rate = 0.03, weight\_decay = 0.01, num\_epochs = 20

device = cuda:0



**SVM图像多分类：**

* 基本原理：将多分类问题转化为若干个二分类问题，对每个二分类问题训练一个SVM模型，最终通过各个模型的预测结果来得出最终分类结果。

SVM分类采用了sklearn内部的SVC分类器，其中Kernel选择了线性核函数，经过多轮测试，只有采用了线性Kernel的版本测试准确率达到了80%。

svm\_clf = svm.SVC(kernel='linear')

该模型与CNN对比，训练和测试时间翻倍，并且CPU占用率很高，原本计划采用 APO 算法搜索最佳的超参数组合，但由于硬件设备限制无法运行出结果，故训练测试参数均采用函数默认值，最终得到测试准确率为81%。



**总结：**

通过使用CNN和SVM算法对数据集进行图像训练分类的过程中可以看出，在处理数据量较大的问题上，神经网络有着明显优势，不仅体现在分类的准确率上，还体现在程序的运行速度和对硬件设备的要求上，但在处理精度上，理论上SVM 比 CNN 更准确，因为 SVM 可以使用线性和非线性分类器来拟合数据，而 CNN 则需要处理高度复杂的非线性数据。但是，CNN可以处理更大的数据集和更复杂的任务，这是SVM无法匹敌的。

**参考文献：**

[1] 图像分类 – GTSRB, fine-tune resnet50, CSDN, 2023年2月22日, <https://blog.csdn.net/lovechris00/article/details/129163478>

[2] chatgpt gpt-3.5-turbo API