实验报告

学号：2017202021

姓名：陈骋

实验说明：

实验平台：车辆：arduino 数据处理：python3.7 + pytorch

实验阶段：通过手机采集车辆前方图像数据，并传送至电脑端进行处理。电脑根据手机发送的图像数据，使得小车能在画好的轨迹上进行正确的运行。

器材：9v干电池\*1，单层两轮小车\*1，arduino uno开发板\*1，L298N电机驱动模块\*1，BT08B蓝牙串口模块\*1，HC-SR04超声波单元\*1，手机\*1，杜邦线若干

实验过程：

1.通过pygame模块实现电脑键盘的监控，同时根据不同的键盘输入对小车进行不同的控制。根据不同的行进方向，通过手机返回的图像信息，存储为不同组别的图片，实现数据的采集。

2.从存储的图片中载入图片信息，使用opencv块实现图片的灰度化和二值化处理，使得道路两侧的标记与道路产生足够的区分度。并使用独热码实现对不同图片的标记。

3.使用pytorch模块，自定义模型。其中此次实验使用的模型为包含两个隐藏层，隐藏层之间的激活函数采用Relu函数，损失函数采用MSELoss函数，优化方法采用SGD函数，并设定相应的学习率。最终经过多次训练得到相应的模型。

注：自定义模型如下：

class MLP(nn.Module):

def \_\_init\_\_(self, input\_size, hidden1\_size, hidden2\_size,output\_size):

super(MLP, self).\_\_init\_\_()

self.fc1 = nn.Linear(input\_size, hidden1\_size)

self.fc2 = nn.Linear(hidden1\_size, hidden2\_size)

self.fc3 = nn.Linear(hidden2\_size, output\_size)

def forward(self, x):

x = F.relu(self.fc1(x))

x = F.relu(self.fc2(x))

x = self.fc3(x)

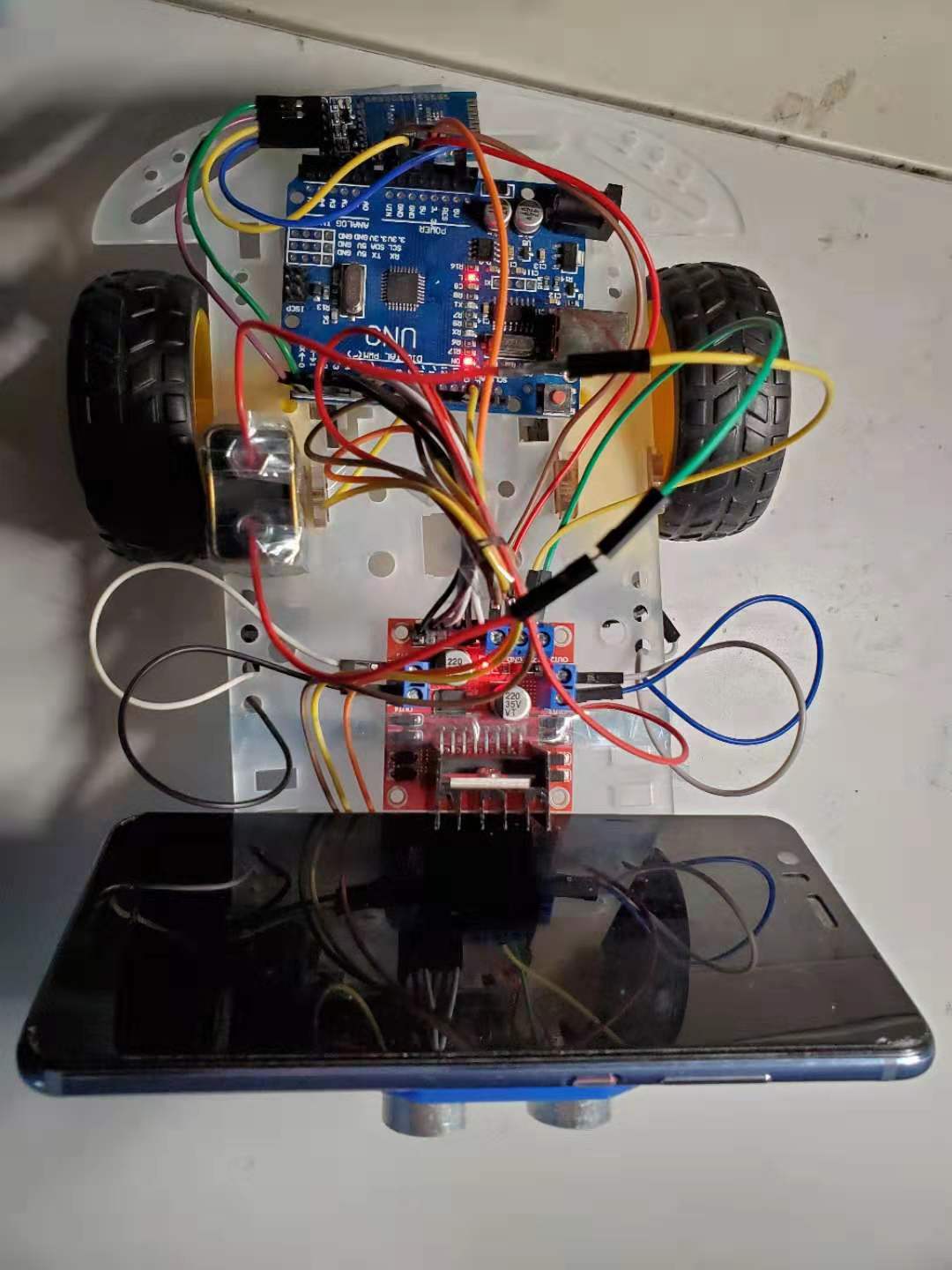
return x

4.模型训练结束后，通过定时的采集手机返回的图像信息，进行与采集数据时相同的图像处理，使用训练完成的模型进行预测，与不同行进方向对应的标签进行比对，得到正确的预测结果，并通过python的串口模块控制小车的运行。

试验中遇到的问题：

在模型定义时采用了以原数据大小作为输入大小，并定义了三层隐藏层。最终在运行时出现了内存爆炸的错误，导致了电脑的崩溃。因此在后期过程中，考虑到实验数据较为简单，因此更改为使用两层隐藏层，并且在收集数据时对图像数据进行一定的压缩。

在模型训练过程中，出现了无法使用交叉熵作为损失函数的错误，经过资料的查询，发现交叉熵函数的数据属于需要满足一定的格式，因此转用pytorch的MSELoss函数作为损失函数

在模型训练过程中出现了数据变为NaN的情况，导致后续数据全变为NaN。后经过检查，发现部分图像信息的特殊性导致了上述问题的发生，因此实际实现时，采取了对所有的图像数据都加一的操作。

小车图片：

小车控制及数据分析代码见/src/scripts/video-control-car/