**送货机器人——第一组第三次报告**

报告人：刘一键、蔡子依、欧阳妍妍、梁景俊、冯晋嘉

1. **小车功能介绍**
2. **应用场景描述**

首先，送货机器人接收到送货指令，得到一个目的地的地址，小车获取与客户的货物提取码，以便在目的地与客户接头；小车根据当前位置预估出到达目的地所需的时间，如果考虑真实的道路交通情况，则需要提前对送货路径做好规划，在本次课程项目中，我们暂不考虑道路情况，设置小车始终以点到点的直线距离送至目的地；当小车上路以后，需要对道路上的状况作出反应，一方面，对前方障碍物做到自动避障的操作，避开障碍物以后需再次调整前进方向，继续向目的地进发，另一方面，小车可识别道路上的红绿灯，做到遵守红灯停、绿灯行这类基本的道路交通规则，为简化项目的难度，我们计划只实现简单的红绿灯识别功能，不考虑复杂的真实道路画面；当小车距目的地在一定范围以内时，可认为小车已抵达，此时便停止行进，等待客户前往核对取货码，当成功与客户接头以后，客户可将小车改为跟踪模式，让小车跟随客户进一步前往具体的目的地，比如客户家中，以帮助客户搬运货物；当客户完成取货操作时，小车进入返回模式，重新规划路线，以最快的速度返回仓库。至此，送货机器人就完成了一份订单。

1. **功能分解**

以下功能模块为项目初期的开发计划，具体的开发方法和成果可能与预期存在差异。

1. **小车普通模式**

在没有订单的时候，小车处于等待状态，此时我们设计小车为手动控制模式，通过手动控制可以将小车推回仓库跑道，与其他送货机器人一起排队，等待送货指令到达。此时的通讯方式为蓝牙通讯。

1. **小车送货模式**

在实际生活中，物流系统接收到用户的订单以后即开始配送，在此次课程项目中，我们通过作为服务器的笔记本电脑，以蓝牙通讯的模式，向小车发出送货指令，并同时发出目的地地址，该地址以经纬度的方式呈现，小车收到指令后，进入送货模式，首先根据当前自己的地址和目的地地址计算出大致的距离，并预估出送货所需的时间，将这两个数据返回服务器，当服务器回送同意出发的指令以后，小车即可开始配送货物。

1. **自动驾驶模式**

小车开始配送货物，正式进入自动驾驶模式，驶入交通道路。在这个模式下，小车将要对实时发生的道路状况进行应对，主要分为以下几种情况：

·**自动避障**：当小车前进过程中，遇到前方有阻碍时，可以通过超声波回声测距模块准确识别到障碍物的存在，并在与障碍物之间的距离达到一定阈值时进行转弯，以避开障碍物。此种避障方法有一个弊端即为无法识别侧方位的障碍物，因为如果超声波的入射角大于九十度，则障碍物会将超声波反射到另一个方向，即当小车驶向侧方位的障碍物时，时无法识别它的存在的，后期我们计划通过图像识别的方法来多方位的识别障碍物，另一种方法是在小车上安装多处超声波测距模块，只有所有测试点都安全以后才可判断为无障碍物。

·**重新定位**：小车每次避障结束以后，方向均发生了一定的变化，但小车仍然需要完成配送货物的指令，因而需要随时对自己的方向和位置进行监督，以便及时回到送货的路径。要实现这部分功能，我们需要小车自行处理GPS模块搜集的地理位置数据，除了位置坐标，很重要的是小车的方向，我们可以通过电子罗盘模块获取小车的方向信息，结合两种模块的数据，即可让小车尽快回到送货轨迹。

·**识别红绿灯**：在道路上行进的时候，为了让小车更有实用性，我们还考虑了道路上红绿灯的情况，让小车实现遵守基本的道路交通规则的功能，即红灯停，绿灯行。我们在开发过程中，想到两个方法来实现这一功能，其一是利用颜色识别模块，它是一种光感元件，通过识别红色和绿色带来的反射光，来判断其真正的颜色，但是此种方法局限性很强，因为感应器只能感受到它正前方的颜色；另一种方法是利用图像识别的方法，但用此种方法需要有抗噪能力很强的模型，因为小车拍摄到的图像范围是很广的，除了红绿灯本身还有许多别的物体，因此我们还需要先识别出红绿灯在图像中的位置，下一步才是识别它的颜色。基于复杂的现实情况，我们最终只能实现简单的红绿灯识别功能，极大地简化了实际情况。

1. **原地等待模式**

我们设置，当小车距离目的地较近时，即可停止行进，具体操作方案为当小车在目的地周围一定范围内时，就停止前进，等待客户前来接头。此事涉及到一个身份验证的问题，我们为了简化问题，计划以验证码的方式让小车进行身份验证，当客户前来取货时，小车会拍摄下客户的照片，回传至服务器，存储起来以便形成该客户的资料库，也可以用于以后更为智能的人脸识别功能。当客户与小车完成对客户的身份验证以后，可听从客户的指令，更改为跟踪模式或者直接到返回模式。

1. **跟踪行走模式**

我们设计此部分功能模块的初衷是，让小车有帮助客户运送货物的功能，当有的货物比较重或是体积比较大，不方便人工搬动时，我们可以让小车帮助客户将货物搬送到更具体的地点，因此此时使用跟踪模式，让小车跟随者客户，前往具体地点。我们在实际操作时，以图像识别的方法，让小车跟踪一个红色物体行走，当然在现实生活中，这样做是不太合理的，我们的想法是后期可以让小车和客户的手机建立起通讯链，而小车的跟踪对象，就变为手机，跟随着信号源行动，该想法目前只是初步的，后期还需要更多通讯方面的知识来验证它的可行性。

1. **自动返回模式**

当小车进入自动返回模式时，即反过来以货物仓库为目的地，规划路线，自动驾驶前行，这与送货时的过程实际上类似，只是要确定好何时进入返回模式。

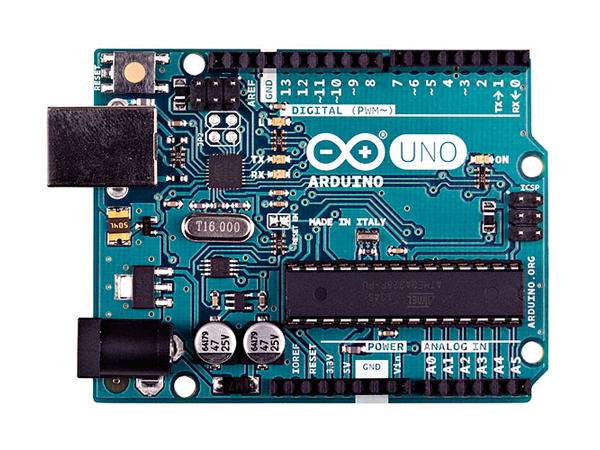
1. **系统架构设计**
2. **硬件配置**

·Arduino uno开发板：XTWduino UNO R3开发板，可以用来做电路连接的Arduino电路板，

Arduino能通过各种各样的传感器来感知环境，通过控制灯光、马达和其他的装置来反馈、

影响环境。板子上的微控制器可以通过Arduino的编程语言来编写程序，编译成二进制文

件，烧录进微控制器。



·超声波回声测距模块：HC-SR04超声波传感器，该模块精度高，盲区只有2cm，探测距离

为2cm-450cm，感应角度小于15度。

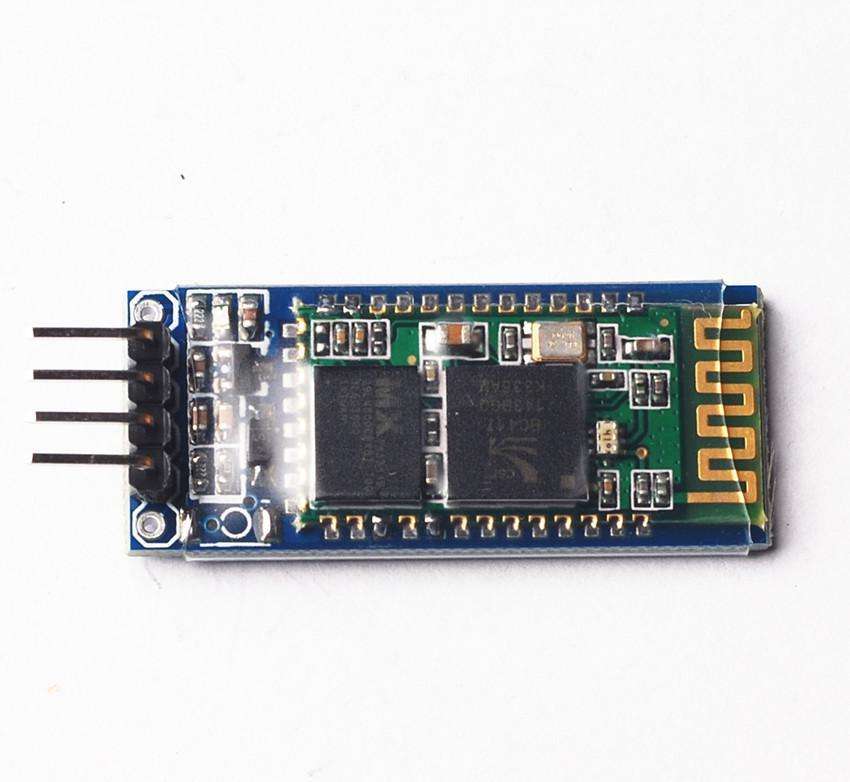


·蓝牙模块：BT06蓝牙串口模块，遵循V3.0蓝牙规范，支持UART接口，支持SPP蓝牙串

口协议，成本低、体积小、功耗低、收发灵敏性高。SPP-CA蓝牙模块串口电平需3.3V，

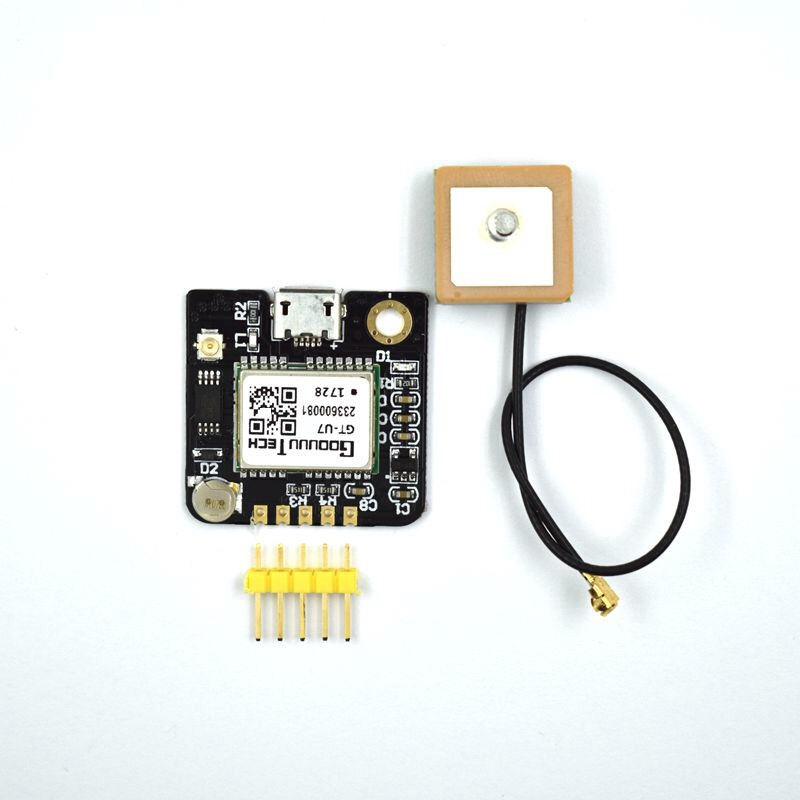
如果和5V电平系统连接，需要增加电平转换芯片。蓝牙信号受周围影响很大，如树木、

金属、墙体等障碍物会对蓝牙信号有一定的吸收或屏蔽。

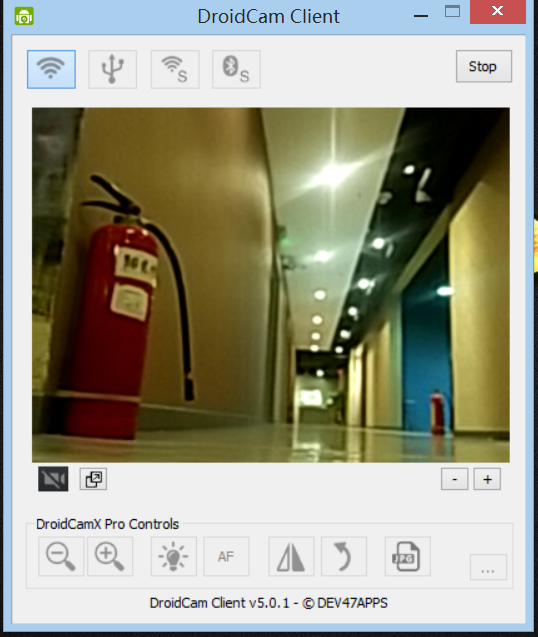


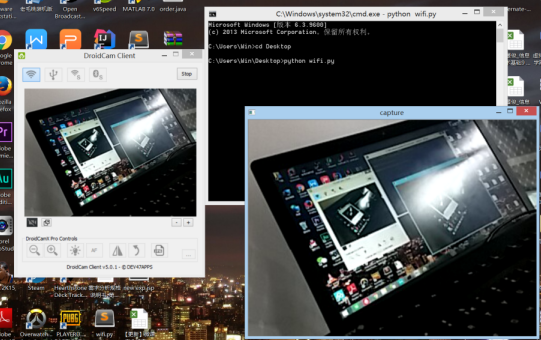
·GPS模块：GT-U7模块，高灵敏度、低功耗、小型化，工作电压3.6V-5V，带IPEX天线接

口，可快速定位，板载E2PROM可保存参数数据。



·摄像手机：安卓手机，安装有DroidCam，与计算机通过wifi连接并通讯。

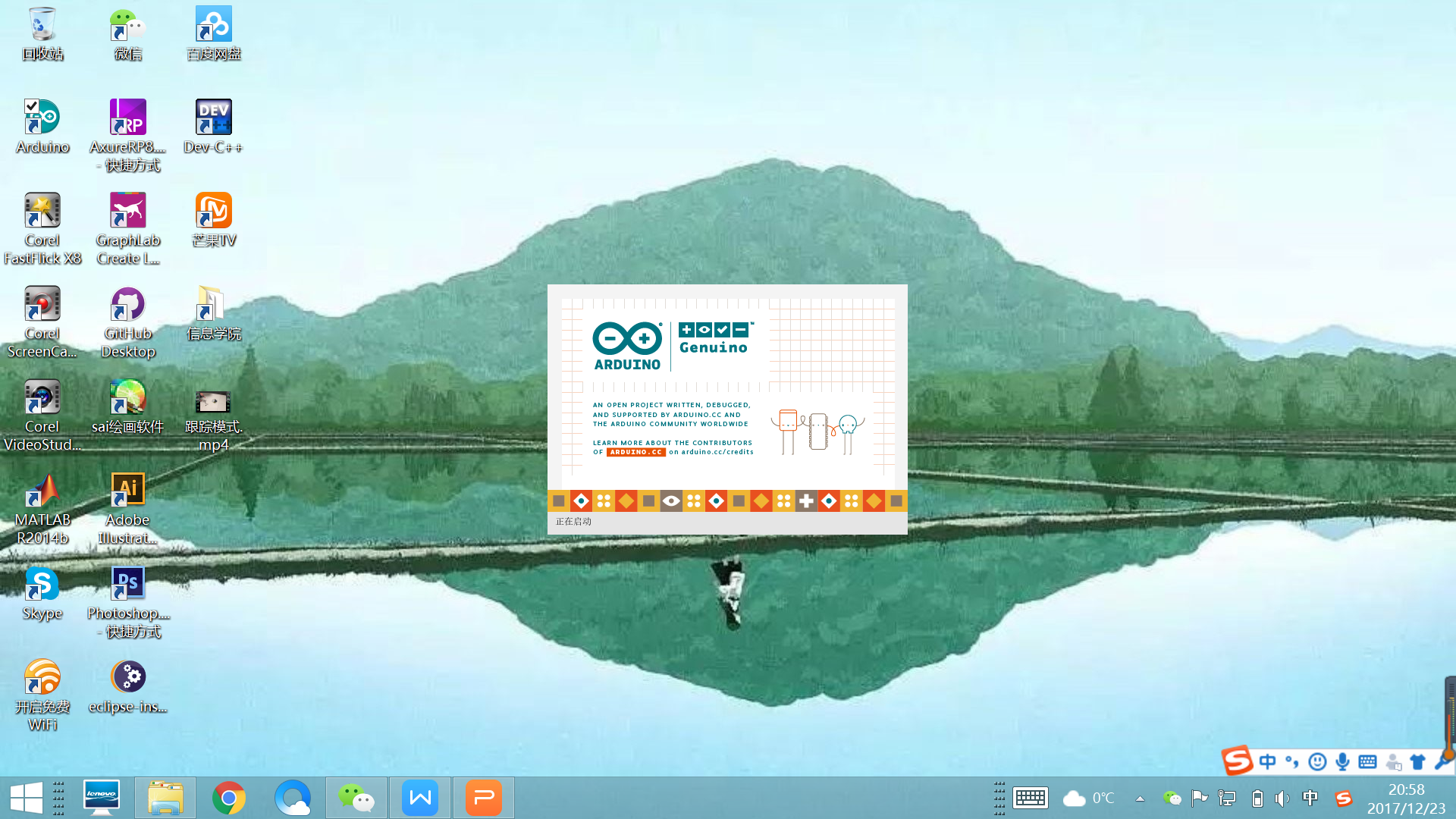




1. **软件**

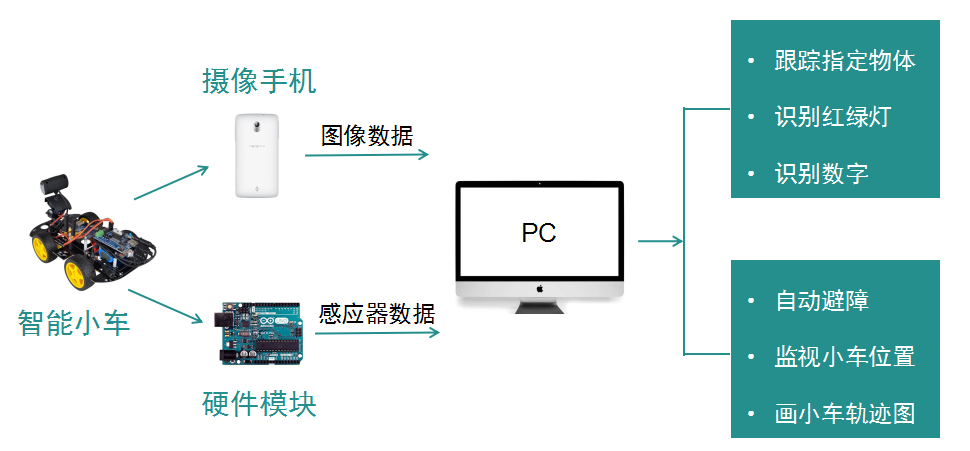
·Arduino开发环境：Arduino IDE，Arduino IDE基于processing IDE开发，可以在Windows、

Macintosh OS X、Linux三大主流操作系统上运行，编程语言可使用C、C++等。

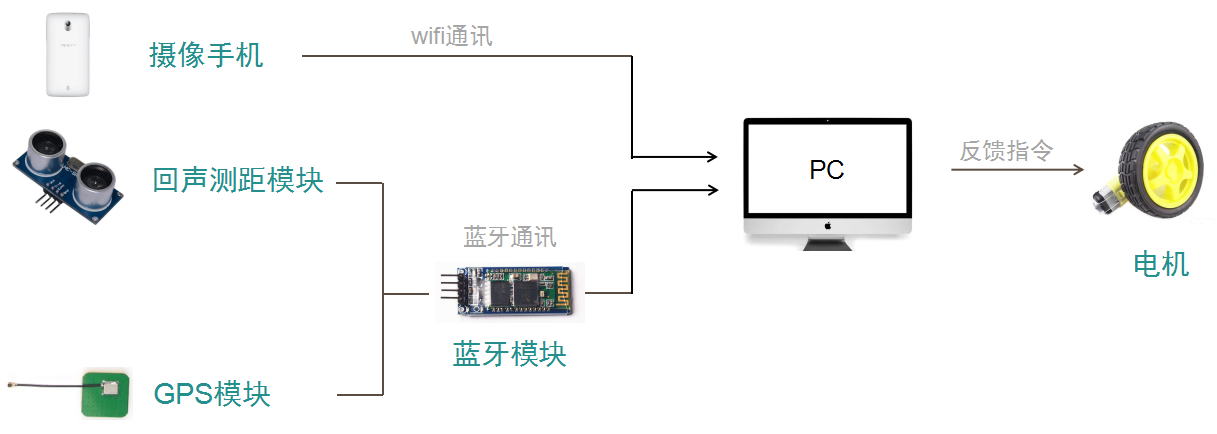


1. **整体架构**

如图所示：



（图一）



（图二）

1. **开发过程 & 调试方法**
2. **蓝牙通讯**

本功能实现小车能够取消自动运行模式，由手机蓝牙遥控小车的转弯、前进和后退，手机发送命令信号，小车接收后反馈消息到手机中。

1. **问题1——小车无法接收控制信号**

原因：误将开发板的输出输入端口RX 、TX端口要分别对应蓝牙模块的RX、TX端口。

解决办法：将开发板的TX端口连接到蓝牙模块的RX端口，开发板的RX对应蓝牙模块

的TX端口。

1. **问题2——小车可以接收控制信号，但不能发送反馈消息到手机**

原因：创建了BT对象作为蓝牙的输入输出端口，使用BT.Read()方法接受蓝牙消息，但

却使用Serial.println()方法发送反馈消息。

解决办法：使用BT.println()方法代替Serial.println()方法，手机成功接收到消息。

1. **物体跟踪**

本功能实现小车自动跟踪某一个颜色的物体，具体实现方法如下：

为小车安装摄像头，摄像头不断抓取小车前方路况视频流

摄像头将抓取的视频流上传到wifi局域网中

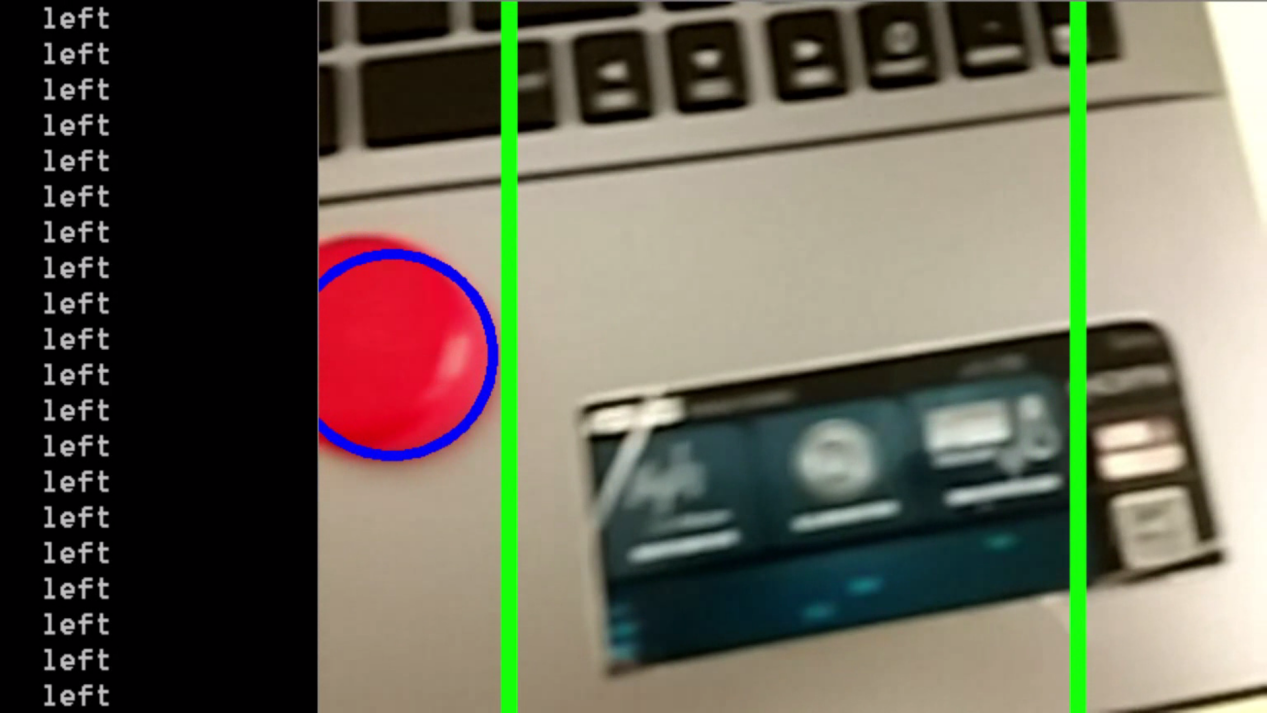
电脑在局域网内抓取视频流并截取图像帧

电脑对图像进行颜色识别，判断目标颜色处于图像的左侧、中部还是右侧

根据颜色块的分布决定小车左转、向前还有右转

电脑通过蓝牙发送目标命令

小车改变行驶方向



检测物体左右位置

**·问题1——电脑无法通过python的bluetooth库连接到小车，显示小车蓝牙未开启**

原因：未知

解决办法：电脑首先通过操作系统GUI界面手动连接小车蓝牙，再查询电脑的串口使用

状态获取与小车蓝牙通信的串口号，使用python的Serial模块方法直接通

过串口与小车蓝牙进行通信。

**·问题2——小车转向后容易丢失颜色对象，导致小车无法继续跟踪**

原因：小车转向速度、幅度过大，而视频流的帧数过低导致不容易紧跟跟踪对象

解决办法：小车转向时减小马达马力，降为原来的一半，并且只允许一个轮胎工作。

此时抓取的图像比较清晰，容易识别颜色

**·问题3——调整小车转向速度后右转不正常，左转正常**

原因：小车右轮转动阻力较大，采用一个马力左转的时候无法正常工作

解决办法：小车左转时采用全马力工作，右转时采用半马力工作。

1. **数字识别**
2. **训练数据采集**

我们希望实现小车能识别数字倒计时的功能，因此搭建一个卷积神经网络进行数字识别。数据集为MNIST手写识别库，包含了近一万张手写数字的图片，每张图片都是28\*28的像素格式。每一张图片都带有一个标签，表示这张图片中对应的数字。

我们搭建的卷积神经网络包含两个卷积层，一个全连接层、一个dropout层和最后的softmax分类器。卷积核的大小为5\*5，maxpooling size为2\*2，第一卷积层映射到了32个通道，第二卷积层映射到了64个通道，因此在全连接层之前，图像处理为7\*7\*64 的张量。

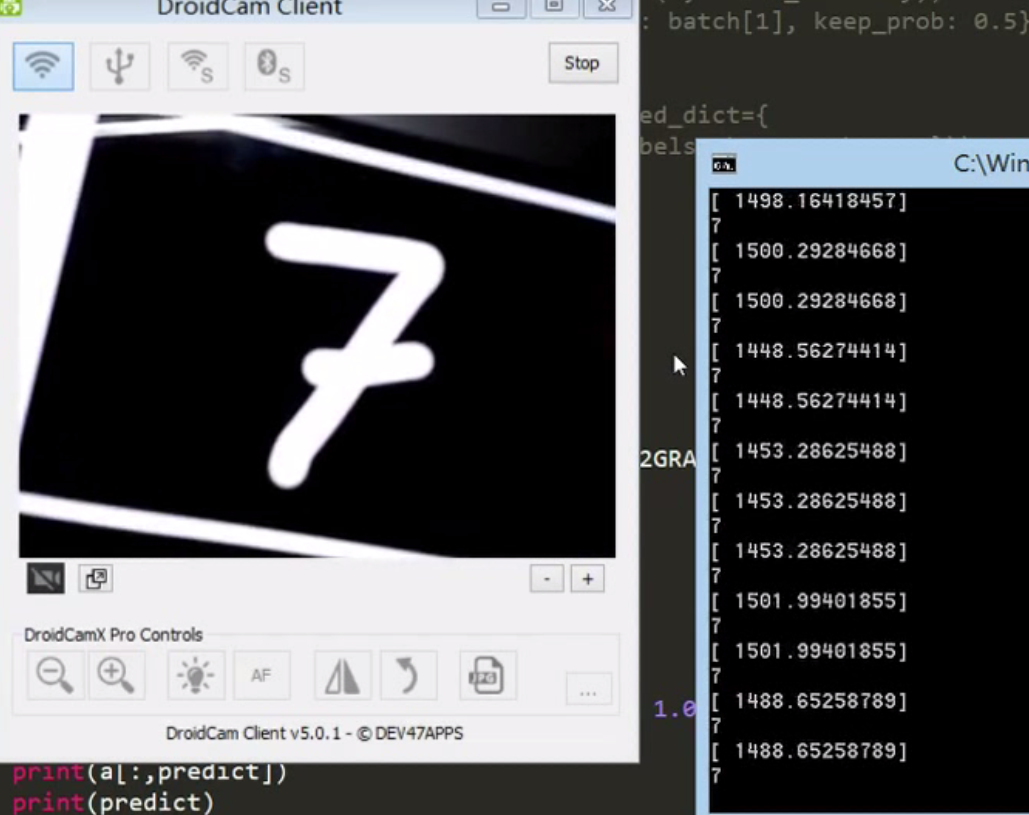
最后通过dropout层和softmax层输出分类结果

在测试阶段，首先使用opencv工具将图像resize成28\*28的图像并进行灰度处理，取其中一个通道，再展开成1\*784的张量，作为网络的输入，最后输出分类结果。

但由于数据集是纯黑色背景，白色线条的数字，不复合实际场景。因此我们以小车视角抓拍了15000张小车前放置数字的图片，并作为数据集重新训练模型。这张的数据集不仅包含数字，并且带有一些杂物的背景，我们希望小车可以消除这些杂物的影响。图片尺寸也从32\*32增加到128\*96。



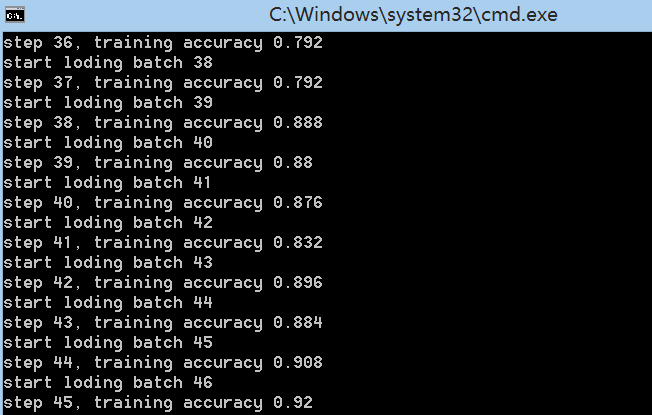
数据集实例



训练结果测试截图

1. **模型过拟合问题**

但是在训练过程中我们发现我们抓取的数据集特征比较集中，导致了模型的泛化能力较弱，具体表现为我们的模型在验证集上的结果非常好，达到了90%以上。但是我们在实体测试的时候发现，模型识别的准确率比较低，出现了明显的过拟合的情况，即使增加了dropout层，也没有太大的改善。原因应该是测试集中频繁出现的背景元素迷惑了模型的调整，这些噪声元素也列入了模型的参考范围。由于数据集的采集比较繁杂，我们最后决定录制训练结果的场地定于之前收集数据集的同样的地方，获得了比较好的效果。



训练及验证结果

1. **GPS定位**

**（1）算法实现**

我们为小车增设了GPS模块，希望实现小车自动到达某个指定的地方。由于GPS模块只具有判定所在位置的功能，不能识别小车的前进方向，因此我们对一般的路径搜寻算法进行的改进，改进之后的算法如下：

输入：目标位置的经纬度，初始化小车和目标位置的最短距离为正无穷

1.接收GPS模块的数据，计算小车当前位置与目标位置距离

2.如果两者距离正在减小，小车则保持方向，继续往前

3.如果两者距离正在增加，则向小车发送转向的指令

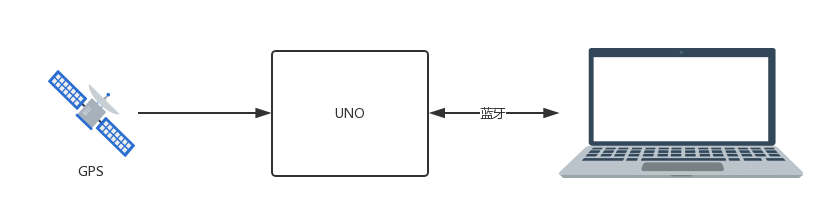
4.如果小车接收到了转向的指令，则随机转动一定方向并前进一小段距离，重新开始发 送GPS位置

5.如果小车和目标位置距离小于一个阈值，则判定小车已到达目的地

6.如果小车中途遇到了障碍物，则随机转向并前进一小段距离，重新开始发送当前GPS 位置通过这种算法，保证了小车总是朝着接近目标位置的方向前进，但是不一定是最优 路径。

**（2）通讯实现**

小车的GPS模块首先通过软串口输入到UNO板中，并通过蓝牙传输到服务器，服务器计算数值之后将指令以蓝牙的方式回传到小车蓝牙模块，并输入到UNO板中，由UNO控制小车的转向或前行。



**（3）实施困难**

当我们组装好小车之后运行发现，小车总是丢失来自蓝牙或者来自GPS一方的信号，导致整个系统没有办法正常地工作。查阅了大量资料之后发现，原来是因为我们的两方面信号都是通过UNO的软串口接收的，但是UNO板没有办法同时缓冲来自多个信道的信息，因此我们需要通过listen()方法指定监听哪一个信道，否则总是以最后一个开启的信道为准。

根据这种限制，我们改进我们的算法：首先开启GPS端口，监听位置的信号，接收到信号之后转发到服务器，并同时监听蓝牙信号，等待服务器的操作指令，接收到操作指令之后，马上开启GPS端口，监听位置信号。

1. **组员分工**
2. 材料采购：蔡子依70%，刘一键20%，冯晋嘉10%
3. 小车搭建：蔡子依50%，刘一键40%，梁景俊10%
4. 代码调试：梁景俊50%，刘一键20%，欧阳妍妍10%，蔡子依10%，冯晋嘉10%
5. 文档撰写：欧阳妍妍40%，刘一键30%，梁景俊20%，蔡子依10%，
6. 课堂汇报：欧阳妍妍50%，梁景俊50%