

第二次实验报告

王诗俊 2015201951 廖钰蕾 2015201953 孟妍廷 2015202009 刘笑 2015201925

2017 年 11 月 4 日

一 目标:

1. 替换上一阶段损坏的电池
2. 利用手机蓝牙控制小车行驶
3. 在小车上固定开启着摄像头的手机，手机通过 WiFi 将摄像头所拍画面实时传至电脑端
4. 电脑端每隔 2s 截一张图，根据自己训练出的模型选择合适的转弯方向
5. 对图像进行处理，使小车能够识别红绿色，“红灯停，绿灯行”

二 收集数据:

1. 数据要求

小车在实际行驶过程中以不同角度靠近墙壁的图片

2. 数据获取

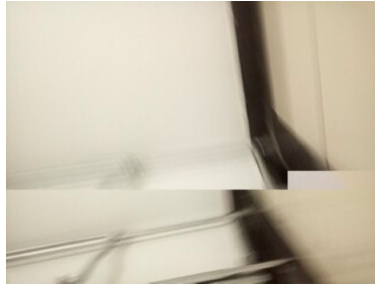
- (1) 电脑端安装 DroidCam 软件，手机端（固定在小车上）安装对应 APP，连在同一局域网
- (2) 手机端打开摄像头，电脑端能看到手机拍摄的实时画面（WiFi 状况良好的情况下）
- (3) 电脑端安装 Python 的 opencv 包，编写 py 程序调用此包，程序功能是每隔 2s 截一张图片并保存
- (4) 人为控制：
 - 手动控制小车撞墙的角度，保证角度多样化
 - 在小车撞墙后，手动将小车与墙壁分离
 - 在小车翻车后，手动将小车扶正
 - 在小车零件被震翻、线路被震断时，手动恢复

3. 时刻关注电脑端的截图、WiFi 连接状况和截图完成情况

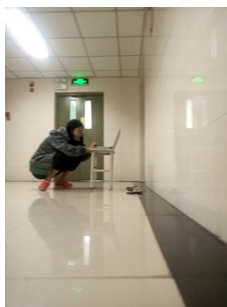
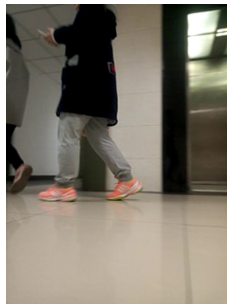
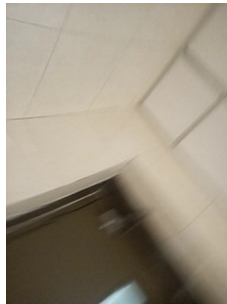
4. 数据筛选

- (1) 剔除了各种“人类无法识别”的图片 (WiFi 刚连接的前几张图片为纯色的、WiFi 连接不稳定时的图片会错位)





(2) 剔除了各种“非撞墙”的图片(天花板、楼道里来来往往的人、地板上的小虫子、蹲在电脑前监控拍摄进展的队友)



(3) 剔除了各种“无法标注”的图片(离墙太远(无需转弯)、离墙太近(整个图片都是白花花/灰惨惨/黑漆漆的墙壁))



5. 数据量

原始数据量：2600 张图片（26 个字母、每个字母 100 张）

筛选后数据量：1787 张图片

三 代码逻辑:

——训练模型

1. 图片预处理

对图片灰度处理，使用高斯平滑处理图片降噪，再使用 Canny 边检测器检测物体边缘

2. 特征提取

用 SIFT(尺度不变特征变换) 来提取特征点，得到 $n \times 128$ 的特征向量 (n 为特征点个数)

3. 向量量化

(1) 利用词袋模型对向量进行聚类，将每一簇向量看成一个“词”，将一幅画看成一个“袋”。

(2) 利用 k-means 算法提取所有图片的 SIFT 特征，若共有 n 个向量，将这 n 个向量划分成 k 类，对每一维 SIFT 特征，均映射到一维上。

(3) 计算每个图片中的每一类向量出现的频率，做归一化处理。

4. 模型调优

采取 SVM 模型，采用 GridSearchCV 函数在参数范围内自动调优，寻找最优超参数，范围如下：

```
parameter_grid = [  
    'kernel': ['linear'], 'class_weight': ['balanced'], 'gamma': [0.01, 0.001], 'C': [0.5, 1, 10, 50],  
    'kernel': ['poly'], 'class_weight': ['balanced'], 'gamma': [0.01, 0.001], 'degree': [2, 3],
```

```
['kernel': ['rbf'], 'class_weight': ['balanced'], 'gamma': [0.01, 0.001], 'C': [0.5, 1, 10, 50],  
]
```

最优模型保存在“svm.model”中

——运行过程

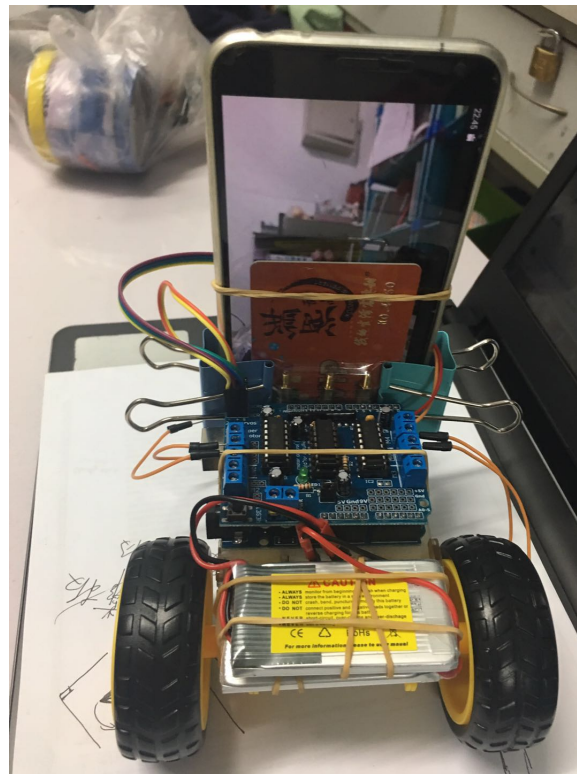
1. 读取模型
2. 实时截取图片，并做红绿判断—将 RGB 图像转化为 HSV 图像，遍历图像判断红绿色彩
3. 发送信号获取传感器距离信息
4. 离墙较近时，对图片进行预处理，特征提取，向量量化等，并使用 SVM 进行分类，决定转弯方向，返回给小车

四 遇到的问题:

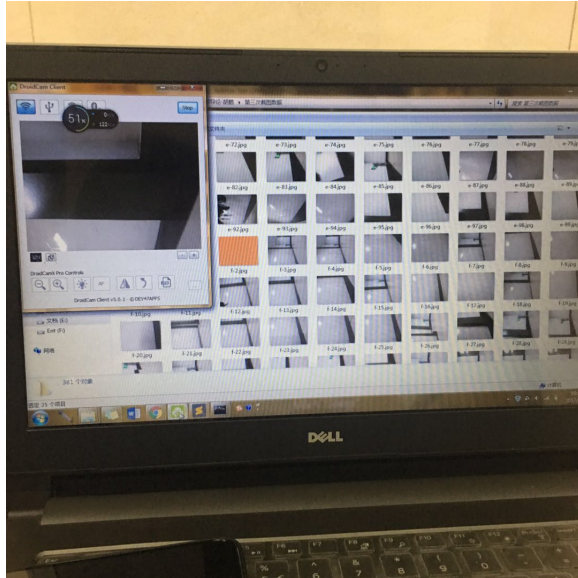
1. 收集数据的效率较低，只能靠人工慢慢控制，数据量较少。
2. 筛选数据时每位成员的标准不同，导致筛选效率不够高
3. 由于 OpenCV 的版本不同，成员在检查代码的时候出现了目前不知道如何调试的错误

五 成果:

1. 小车



2. 收集数据



3. 训练结果

```
##### Searching optimal hyperparameters #####

Highest scoring parameter set:
{'C': 10, 'gamma': 0.01, 'class_weight': 'balanced', 'kernel': 'linear'}

#####

Classifier performance on training dataset

      precision    recall  f1-score   support

 Class-0       0.70      0.62      0.66       283
 Class-1       0.64      0.72      0.68       266

 avg / total       0.67      0.67      0.67       549

#####

#####

Classifier performance on testing dataset

      precision    recall  f1-score   support

 Class-0       0.71      0.66      0.68       96
 Class-1       0.65      0.70      0.68       88

 avg / total       0.68      0.68      0.68      184

#####

#####

lxl@lxl-Lenovo-B40-45:~/aicar$
```

4. 不同 SVM 得到的结果表

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
train_id																											

5. 小车行驶见视频

六 模型评价:

1. 图片预处理: 用 Canny 处理过的图片能明显看到物体边, 且降噪效果相对拉普拉斯等边检测器好
2. 每个图片的特征点个数不同, 而 SIFT 是 CV 中常用的特征之一
3. 向量量化: 在计算机视觉中, SIFT 向量维度较高, 运算速度慢, 可以在此对向量进行聚类, 而词袋模型最初用于文本分析领域, 将每个文档看成袋子, 根据袋中词语进行分类。又由于每幅图对应多个向量, 且个数不定, 所以需要归一化处理。

七 分工:

王诗俊	廖钰蕾	孟妍廷	刘笑
<ol style="list-style-type: none">1. 解决第一次展示中的问题, 寻觅并购买新电池2. 组装小车、固定手机3. 安装软件实现手机端拍摄图片实时传至电脑端4. 编写截图所用的程序5. 完成2600张原始图片数据的采集6. 完成1700+张可用图片数据的筛选7. 编写主函数与小车通信功能的代码8. 拍摄展示视频9. 修改展示用PPT10. 课堂展示	<ol style="list-style-type: none">1. 解决第一次展示中的问题, 寻觅并购买新电池2. 安装软件实现手机端拍摄图片实时传至电脑端3. 查找有关算法的文献4. 选择合适算法5. 编写训练模型的程序6. 编写主函数判断小车转弯方向的程序7. 拍摄展示视频	<ol style="list-style-type: none">1. 组装小车、固定手机2. 完成2600张原始图片数据的采集3. 完成1700+张可用图片数据的筛选4. 制作展示用PPT初稿5. 制作成果报告6. 检查代码7. 剪辑展示视频	<ol style="list-style-type: none">1. 提出本阶段功能目标2. 查找有关算法的文献3. 选择合适的算法4. 完成1700+张图片的标注5. 拍摄展示视频

八 参考文献:

Canny 参考: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/canny.html>

SIFT 参考: <https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>

k-means 参考: <http://www.onmyphd.com/?p=k-means.clustering&ckattempt=1>