外卖员门禁辅助识别系统

设计软件基础　　课程汇报

目录

1 实现背景
空间行为现状 & 设计流程实现

2 模型训练
数据 - 处理 - 训练

3 图像预测
预测 - 状态 - 可视化

4 硬件交互
摄像 - 检测 - 交互

5 视频展示

空间行为现状

H栋一楼各门禁设有权限，而师生常位于2-4层办公学习，不能及时收到送达的外卖和快递，或是被催促取件/取餐。

我们希望通过外卖员门禁辅助识别系统，让外卖员能够以临时权限进入H楼送达外卖，既能便捷师生自由取餐，也能帮助外卖员减少配送等待时间，加快效率。

1 实现背景
空间行为现状 & 设计流程实现

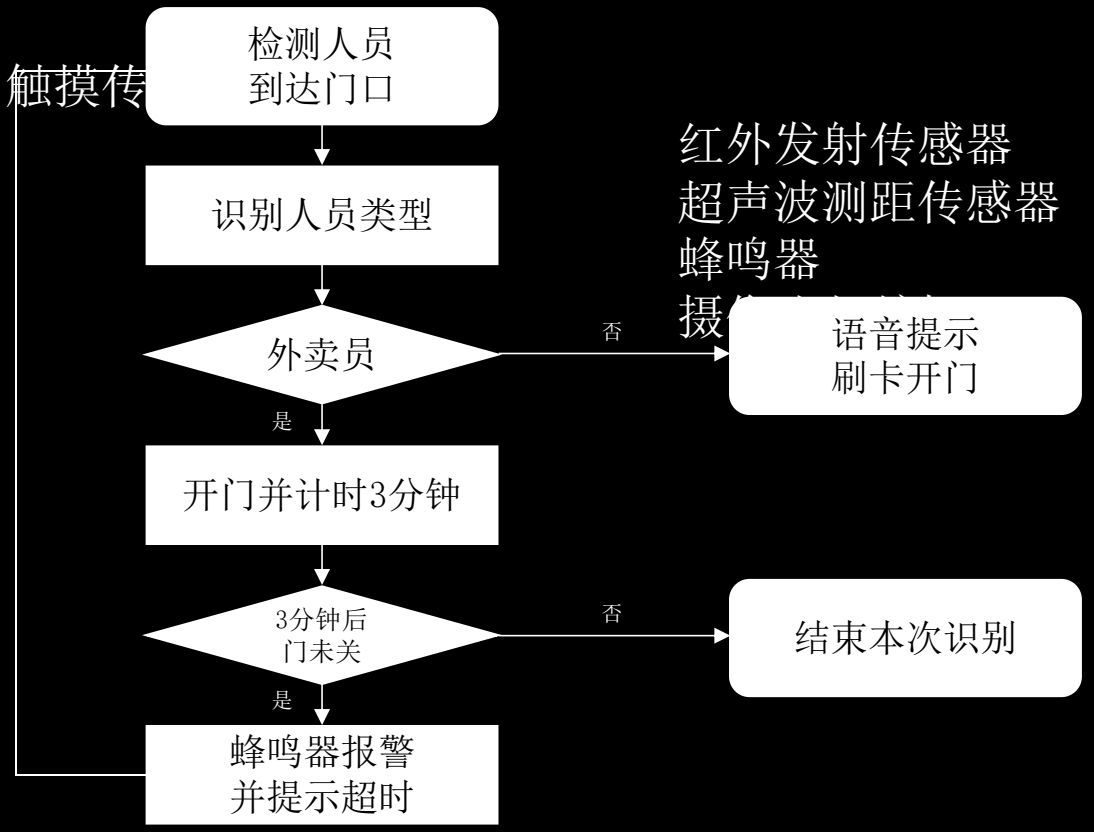
2 模型训练
数据 - 处理 - 训练

3 图像预测
预测 - 状态 - 可视化

4 硬件交互
摄像 - 检测 - 交互

5 视频展示

设计流程实现



红外发射传感器
超声波测距传感器
蜂鸣器
摄

1 实现背景
空间行为现状 & 设计流程实现

2 模型训练
数据 - 处理 - 训练

3 图像预测
预测 - 状态 - 可视化

4 硬件交互
摄像 - 检测 - 交互

5 视频展示

模型训练——数据爬取 *image_crawling.py*

互联网图像数据爬取
800张 外卖员（快递员）图像
美团 / 饿了么 / 顺丰 / 朴朴 / 肯德基 / 麦当劳 / 盒马



550张 非外卖员图像
老人 / 小孩 / 青年 / 中年

1 实现背景
空间行为现状 & 设计流程实现

2 模型训练
数据 - 处理 - 训练

3 图像预测
预测 - 状态 - 可视化

4 硬件交互
摄像 - 检测 - 交互

5 视频展示

模型训练——数据预处理 *data_pretreatment.py*

图像数据分类
划分8:2训练集测试集

模型训练 *Resnet34_train.py*

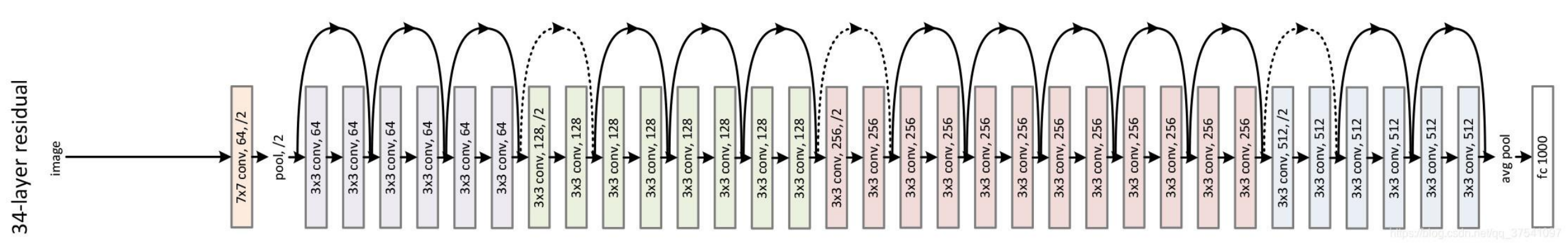
模型训练　*Resnet34_train.py*

二分类问题　基于Resnet34网络进行模型训练

相比于传统的卷积神经网络可以避免：
- 梯度消失/梯度爆炸——通过残差结构解决
- 退化问题——使用BN (Batch Normalization)层解决

由于样本规模较小，额外引入：
- Dropout正则化
- 早停机制
- K折交叉验证 K = 5

以防止过拟合



模型训练 *deliverer_model_final.pth*

```
Fold 1/5
Epoch [1/10], Loss: 0.2408, Accuracy: 0.8936
Validation Loss: 0.0790, Validation Accuracy: 0.9585
Epoch [2/10], Loss: 0.0361, Accuracy: 0.9954
Validation Loss: 0.0749, Validation Accuracy: 0.9677
Epoch [3/10], Loss: 0.0288, Accuracy: 0.9977
Validation Loss: 0.0890, Validation Accuracy: 0.9631
Epoch [4/10], Loss: 0.0641, Accuracy: 0.9977
Validation Loss: 0.0880, Validation Accuracy: 0.9677
Epoch [5/10], Loss: 0.0105, Accuracy: 0.9988
Validation Loss: 0.1143, Validation Accuracy: 0.9585
Early stopping triggered. No improvement for 3 epochs.

Fold 2/5
Epoch [1/10], Loss: 0.3296, Accuracy: 0.8543
Validation Loss: 0.1666, Validation Accuracy: 0.9309
Epoch [2/10], Loss: 0.0661, Accuracy: 0.9884
Validation Loss: 0.1186, Validation Accuracy: 0.9631
Epoch [3/10], Loss: 0.0573, Accuracy: 0.9942
Validation Loss: 0.1141, Validation Accuracy: 0.9539
Epoch [4/10], Loss: 0.0407, Accuracy: 0.9988
Validation Loss: 0.0926, Validation Accuracy: 0.9631
Epoch [5/10], Loss: 0.0226, Accuracy: 1.0000
Validation Loss: 0.0908, Validation Accuracy: 0.9631
Early stopping triggered. No improvement for 3 epochs.

Fold 3/5
Epoch [1/10], Loss: 0.2893, Accuracy: 0.8845
Validation Loss: 0.1074, Validation Accuracy: 0.9537
Epoch [2/10], Loss: 0.1446, Accuracy: 0.9827
Validation Loss: 0.0928, Validation Accuracy: 0.9630
Epoch [3/10], Loss: 0.0612, Accuracy: 0.9815
Validation Loss: 0.0899, Validation Accuracy: 0.9583
Epoch [4/10], Loss: 0.0198, Accuracy: 0.9965
Validation Loss: 0.0832, Validation Accuracy: 0.9630
Epoch [5/10], Loss: 0.0065, Accuracy: 1.0000
Validation Loss: 0.0806, Validation Accuracy: 0.9630
Early stopping triggered. No improvement for 3 epochs.

Fold 4/5
Epoch [1/10], Loss: 0.2968, Accuracy: 0.8822
Validation Loss: 0.0682, Validation Accuracy: 0.9722
Epoch [2/10], Loss: 0.0680, Accuracy: 0.9873
Validation Loss: 0.0727, Validation Accuracy: 0.9676
Epoch [3/10], Loss: 0.0706, Accuracy: 0.9885
Validation Loss: 0.0883, Validation Accuracy: 0.9676
Epoch [4/10], Loss: 0.1728, Accuracy: 0.9861
Validation Loss: 0.1393, Validation Accuracy: 0.9306
Early stopping triggered. No improvement for 3 epochs.

Fold 5/5
Epoch [1/10], Loss: 0.2306, Accuracy: 0.8995
Validation Loss: 0.1077, Validation Accuracy: 0.9630
Epoch [2/10], Loss: 0.0473, Accuracy: 0.9954
Validation Loss: 0.0994, Validation Accuracy: 0.9722
Epoch [3/10], Loss: 0.0813, Accuracy: 0.9977
Validation Loss: 0.1037, Validation Accuracy: 0.9676
Epoch [4/10], Loss: 0.0569, Accuracy: 0.9861
Validation Loss: 0.1220, Validation Accuracy: 0.9491
Epoch [5/10], Loss: 0.0385, Accuracy: 0.9896
Validation Loss: 0.1012, Validation Accuracy: 0.9491
Early stopping triggered. No improvement for 3 epochs.
```

Validation Loss $\approx$ 0.1

Validation Accuracy $>$ 95%

模型训练　*deliverer_model_final.pth*

Tensorboard 训练可视化

1 实现背景
空间行为现状 & 设计流程实现

2 模型训练
数据 - 处理 - 训练

3 图像预测
预测 - 状态 - 可视化

4 硬件交互
摄像 - 检测 - 交互

5 视频展示

图像预测——预测并赋予状态 *Resnet34_prediction.py*

- 检测Screenshots文件夹内图像数量 > 1 触发预测模块
- 利用模型预测图像类别（label）
- 创建状态文本status.txt并根据label赋值文本内容
- 对预测结果进行可视化
- 删除图像并更新status.txt内容为数字2

1 实现背景
空间行为现状 & 设计流程实现

2 模型训练
数据 - 处理 - 训练

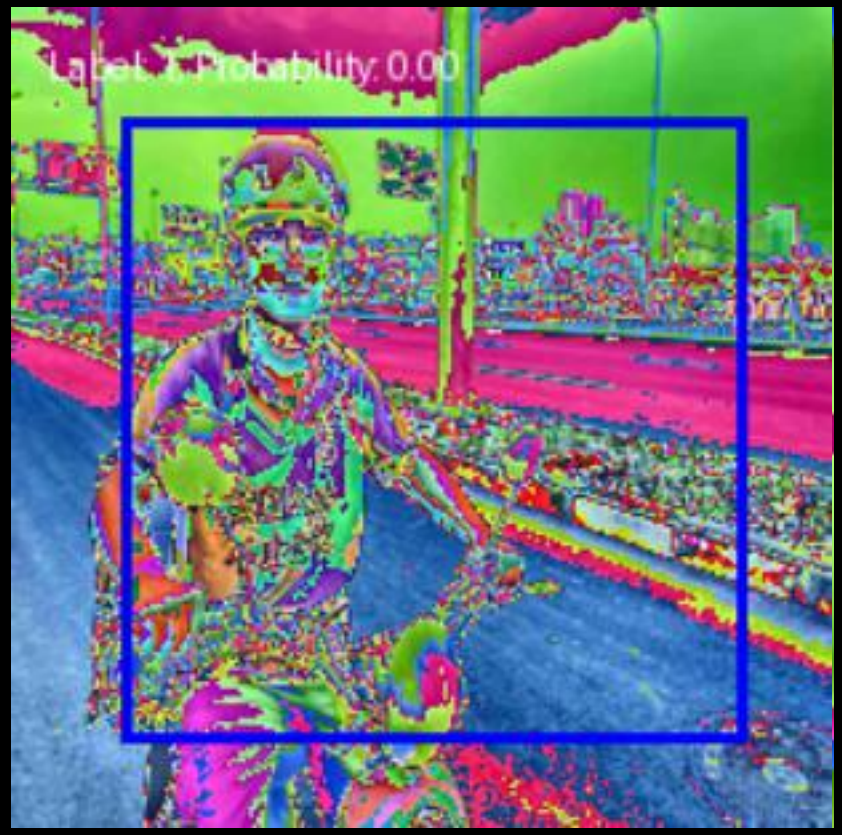
3 图像预测
预测 - 状态 - 可视化

4 硬件交互
摄像 - 检测 - 交互

5 视频展示

图像预测——可视化

- 预测结果标注



1 实现背景
空间行为现状 & 设计流程实现

2 模型训练
数据 - 处理 - 训练

3 图像预测
预测 - 状态 - 可视化

4 硬件交互
摄像 - 检测 - 交互

5 视频展示

图像预测——可视化

- 热图：模型对图像中不同区域的关注程度
  观察哪些部分的特征对最终决策影响最大

1 实现背景
空间行为现状 & 设计流程实现

2 模型训练
数据 - 处理 - 训练

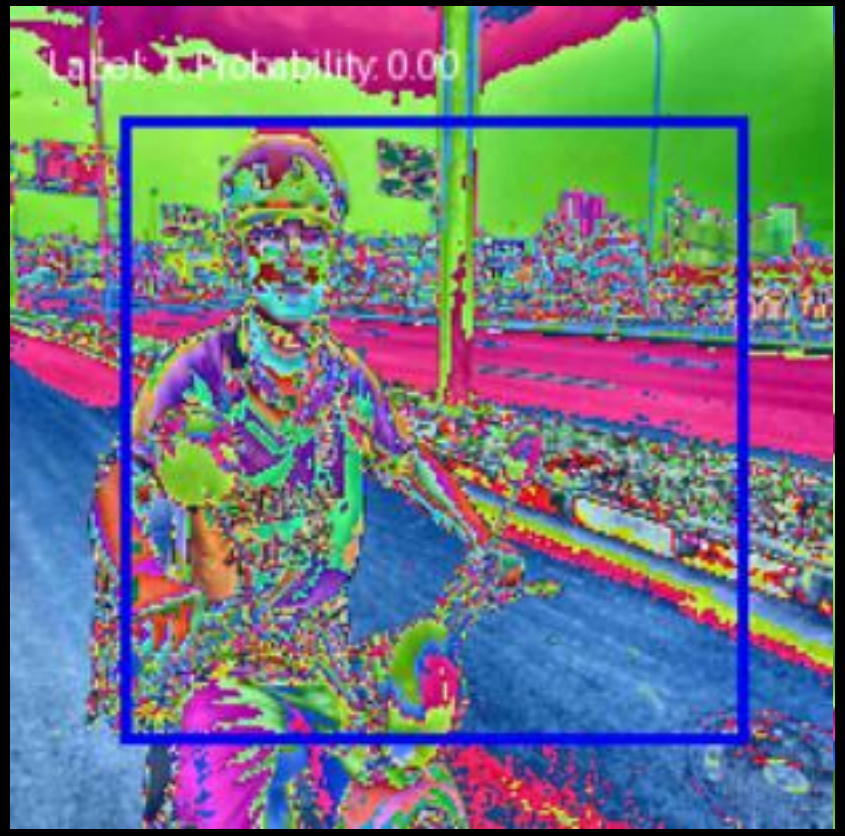
3 图像预测
预测 - 状态 - 可视化

4 硬件交互
摄像 - 检测 - 交互

5 视频展示

图像预测——可视化

- 边界框：显示出模型识别到的对象，包括类别和位置



1 实现背景
空间行为现状 & 设计流程实现

2 模型训练
数据 - 处理 - 训练

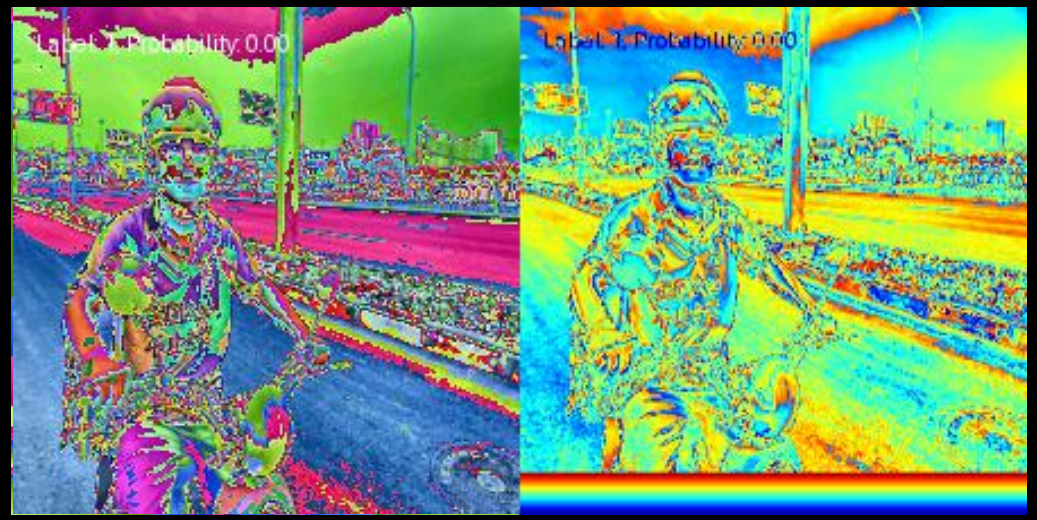
3 图像预测
预测 - 状态 - 可视化

4 硬件交互
摄像 - 检测 - 交互

5 视频展示

图像预测——可视化

- 对比图：分析图像预处理和增强操作的效果



1 实现背景
空间行为现状 & 设计流程实现

2 模型训练
数据 - 处理 - 训练

3 图像预测
预测 - 状态 - 可视化

4 硬件交互
摄像 - 检测 - 交互

5 视频展示

硬件交互——触摸与摄像头 *check_run.py* *main.ino*

- 触摸传感器信号
- 检测到 TOUCHED 字符串开始截图
- 摄像头模块实时捕捉图像
- 保存至screenshots文件夹路径

1 实现背景
空间行为现状 & 设计流程实现

2 模型训练
数据 - 处理 - 训练

3 图像预测
预测 - 状态 - 可视化

4 硬件交互
摄像 - 检测 - 交互

5 视频展示

硬件交互——人员检测与开门 *check_run.py main.ino*

- 触发图像预测模块
- 根据预测结果判断开门情况
- 红外发射器控制舵机开门 / 不开门

1 实现背景
空间行为现状 & 设计流程实现

2 模型训练
数据 - 处理 - 训练

3 图像预测
预测 - 状态 - 可视化

4 硬件交互
摄像 - 检测 - 交互

5 视频展示

硬件交互——门状态语音交互 *check_run.py* *main.ino*

- 外卖员——输出语音提示：
  检测到外卖员，门已开，请在三分钟内关门离开
- 非外卖员——输出语音提示：
  未检测到外卖员，请刷卡开门
- 超声波测距器检测门状态，若OPEN状态超过3分钟则输出语音提示：门超时未关，请立即关门离开
  同时蜂鸣器发出警报

1 实现背景

空间行为现状 & 设计流程实现

2 模型训练

数据 - 处理 - 训练

3 图像预测

预测 - 状态 - 可视化

4 硬件交互

摄像 - 检测 - 交互

5 视频展示