

2022.12.6 - 经验分享会

0x00 - 重点

- 选题选**人多**的更好，因为基数大，晋级机会大。
- 第一组的**产品介绍（答辩）方法**可以参考。
- 可以多看**学位论文**，确定算法、方案、创新点等。
- 多与指导老师沟通。

0x01 - A02 安全驾驶监测系统

1. 角色建议

本比赛服务企业，面向企业，
主要考虑：项目如何落地？产品是否可行？用户是否满意？

人员主要分成三个方面：

- 技术开发 - 前后端开发工程师、算法工程师
- 产品设计 - 产品经理
- 商业策划 - 市场运营

“产品设计”向“技术开发”提供**意见反馈**；“技术开发”向“产品设计”提供产品迭代；
“产品设计”和“商业策划”进行需求且同。

2. 选题建议

- **队伍数较多**的赛题可能晋级决赛**机会更大**（基数大）；反之队伍数更少的需要更出彩。
- 答辩分组并不是一个题目一组，为**横向比较**。若选队伍少的没有比较对象，只能与其他队伍比较。

【所以是选人越多的组越好？我还以为选人多的竞争更激烈……】

3. ★产品介绍（答辩）方法

1. 项目背景的概述

01 项目概述

疲劳驾驶问题不容小觑

疲劳驾驶交通事故占交通事故总量 **21%**
交通事故死亡率高达 **83%**

当前现状

酒驾问题管理复杂

我国每年死于酒驾高达 **10万** 左右
30% 的事故是由酒后开车引起
驾驶员死亡中 **59%** 与酒后驾车有关

交互形式单一

目前汽车驾驶系统交互形式以触控为主，**交互形式少且不友好**

智能安全驾驶监测系统

汽车驾驶天气复杂

不良天气条件是引发交通事故的重要因素
气象原因占交通事故 **25.7%**
高速公路事故死亡中 **70%** 与灾难性天气有关

智能驾驶已成时代趋势

2020年乘用车辅助驾驶渗透率高达 **30%**
预计2025年渗透率或超 **60%**

2. 团队整体功能概述

- ## 02 解决方式



多样化反馈

酒精传感器、摄像头、可穿戴设备、AI开发板等都会根据**不同情况信息进行实时反馈**。



完善的安全监管系统

整合移动端、后台与硬件系统，形成一体化多功能的安全监管系统。从**低头，闭眼，等八大异常驾驶行为**展开，实时获取驾驶员行为、生理特征等数据。



人性化交互设计

为驾驶人提供**手势交互服务和HUD抬头显示交互**方便驾驶过程中用户的信息查看。



Word cloud containing terms: 健康监测与报表应用, HUD交互应用, 驾驶行为监测系统, 天气安全管理系统, 酒精浓度监测系统, 酒驾交互应用, 健康监测与报表应用, HUD交互应用, 驾驶行为监测系统, 天气安全管理系统, 酒精浓度监测系统, 酒驾交互应用.

- 分模块系统讲解，哪个模块完成什么功能

02 解决方案

五大系统



酒精浓度监测系统

- 主动监测**主驾驶**位酒精浓度
- 主动**警报**
- 主动**拨号**



驾驶员行为监控系统

- 八大驾驶行为监测
- 驾驶建议
- 驾驶报表



健康监测与报表应用

- 健康状态**查询与警报**
- 健康报表应用
- 健康建议



个性化安全交互应用

- HUD个性语音交互
- **家属**个性化语音包
- IP形象设计与贴纸



车外环境与驾驶安全管理

- 车外天气
- 驾驶注意事项

3. 产品介绍

1. 架构：系统架构、业务架构、技术架构等



2. 支撑文献



所有的基准判断需要有理有据。

3. 算法介绍

产品介绍

引入多模态的疲劳判断理论

《[PERCLOS: A Valid Psychophysiological Measure of Alertness as Assessed by Psychomotor Vigilance](#)》
PERCLOS疲劳模型，P80判断基准，15帧为周期进行疲劳值的计算

多模态	监测判断指标
行为信号	15帧内的低头、闭眼、打哈欠行为占比
生理信号	心率、血氧饱和度数值大小与波动情况

$$Perclos = \text{Max}(\frac{Rollhead}{Roll} + \frac{Rolleye}{Roll} + 0.4 + \frac{Rollmouth}{Roll} + 0.2, Rollheartsc * 0.25 + RollSpO2sc * 0.4)$$

状态 = { 疲劳 $Perclos > 0.64$
健康 $Perclos \leq 0.64$ }

大致介绍算法，以及该算法怎么应用到产品上。

4. 机器、深度学习部分

03
产品介绍

YOLOv5目标识别算法

YOLO是一种高效的目標检测方法，可以做到实时目标检测。
本项目选择YOLOv5s作为网络结构进行训练，便于边缘计算搭载。

算法	优点	缺点
YOLOv3	借鉴残差网络，实现多尺度检测	模型复杂，对中、大目标检测效果较差
YOLOv4	在检测精度与速度之间取得了权衡	精测精度有待提高
YOLOv5	模型尺寸小，部署成本低，灵活性高，检测速度高	性能有待提高

上游任务

下游任务

Simsiam自监督学习算法

YOLOv5目标检测算法

上下游任务分配

11.03 / 11.11.18

2x

选用模型的原因（与其他模型比较）；训练、测试集的大小、准确率；最终的模型大小和响应时延等指标。

5. 创新点介绍

03
产品介绍

方案介绍

驾驶安全助手

HUD抬头显示

呼叫

状态检测报警

数据查看

天气驾驶安全

接通

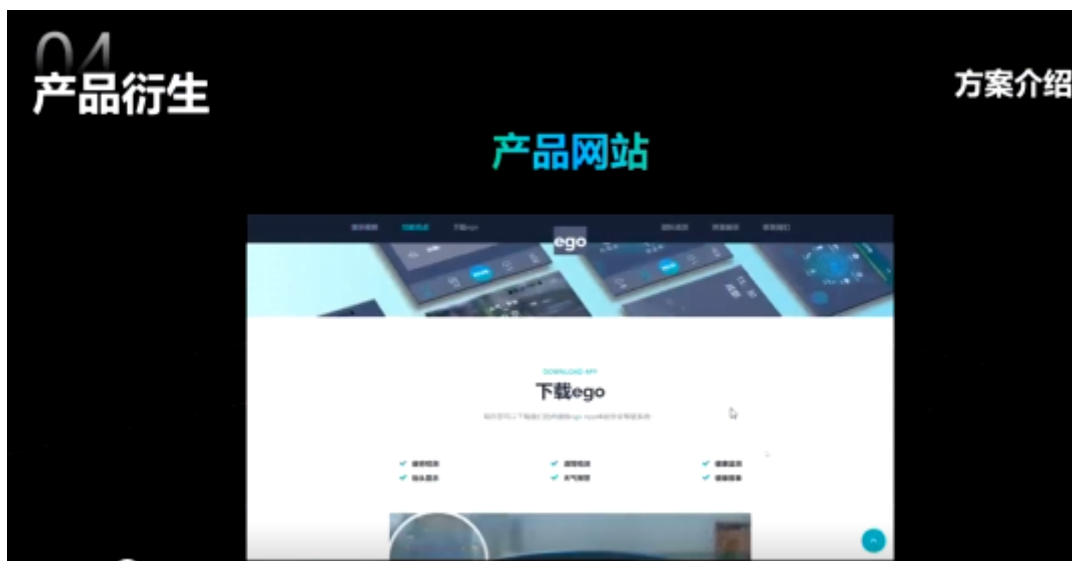
11.22 / 11.11.18

2x

介绍没有要求的创新点。

4. 产品衍生

如：包装、海报、宣传、网站



5. 发展规划



这个产品未来的发展方向，介绍可以少，但要完善和全面。

3. 产品包装（设计人员相关）

设计需要具备的能力：

- PPT展示相关：PPT、PS、海报、排版、视觉设计
- 视频展示相关：PR、AE等剪辑软件
- 软件原型设计：Axure、Figma、MasterGo
- 三维建模设计（视赛题需要）

设计的学习路线：

- 技术类
 1. 熟练使用PS/PPT
 2. 一周时间学会原型设计软件
 3. 一周时间学会PR
 4. 一到两周学会AE基础操作和模板嵌套
- 产品类
 1. 用户需求调研
 2. 需求与功能转化
 3. 产品创意表达
 4. 其他软件学习

包装（撰写文档、答辩）技巧：

- 突出产品技术的重难点、突破点；突出功能特色
- 积极应对突发情况，对预计会遇到的问题进行充分准备
- 要对技术路线熟悉
- 经常联系指导老师

0x02 - A08 缺陷检测系统（图像识别）

1. 如何做好比赛

- 了解**队伍优势**
- 队内
 - 沟通交流（期末后每周开一次组会，讨论进度、问题、与指导老师咨询、后续工作，返校后即实现了所有需求）

- 落实执行
- 赛题
 - 做好**产品定位**（如他们组的就是检测精度），将其**放在开发首位**
 - 亮点功能（查询论文，发现其他人还没实现的功能）
 - 功能取舍（取亮点特色，舍去没技术含量很容易实现的小功能）

2. 图像识别题的流程

- 数据处理



- 补充数据
- 划分测试训练集
- 数据增强（翻转、随机明暗度调整、裁剪填充等）
- 模型调参

模型调参

SSDLite算法设定了与anchor机制类似的**先验框** (Default box)机制。

先验框的宽高比以及尺寸由配置文件参数`aspect_ratios`, `max_ratio`, `min_ratio`决定

K-means输出结果为:

K anchors:

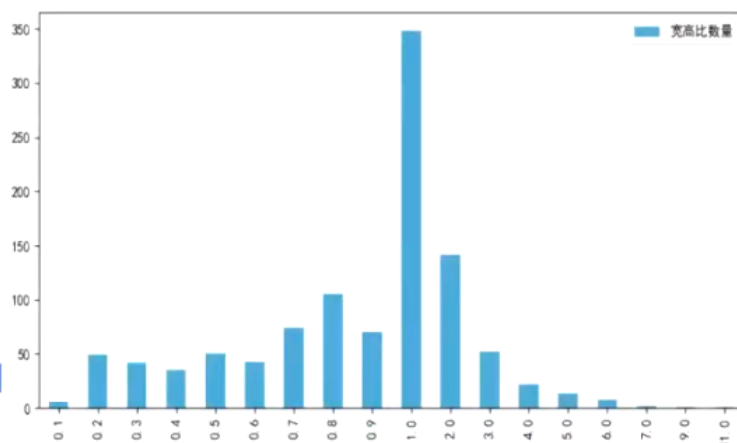
[[31 36][60 92]

[84 256][90 71]

[247 63][264 135]]

Ratios:

[0.33, 0.65, 0.86, 1.27, 1.96, 3.92]



经过对数据集的分析, 我们调整了模型默认参数, 模型**mAP**从**84.7**提升至**86.7**

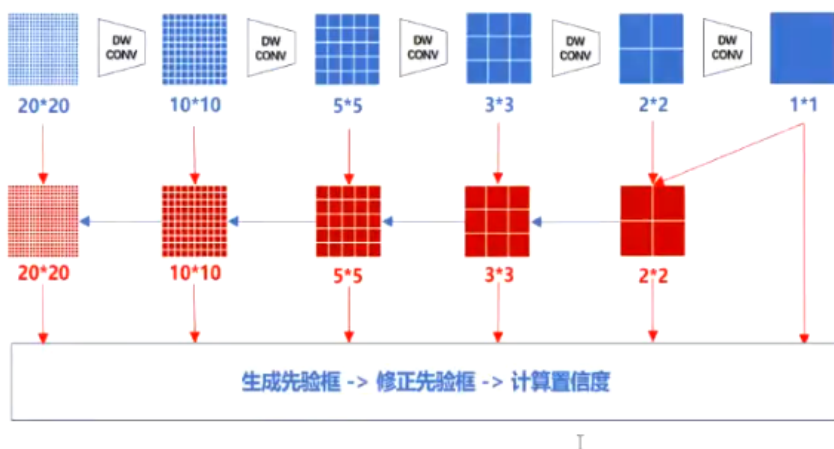
在调整的过程中不断展现mAP (模型精度) 的变化。

- 模型网络 (算法) 改进

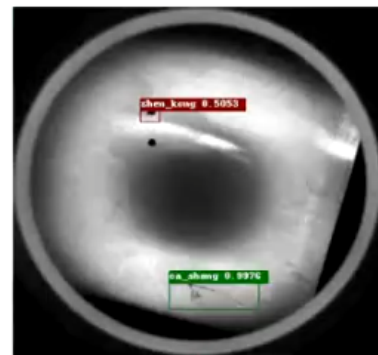
算法改进 - 提升小目标检测能力

SSDLite-Mobilenetv3-large小目标检测不理想

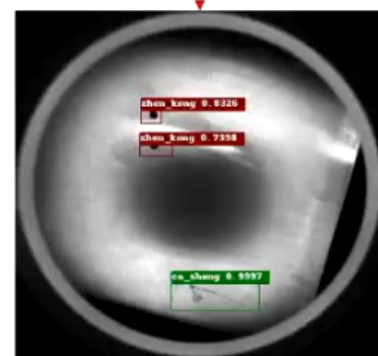
为解决这一问题，我们团队对算法结构进行了改进，在对特征图进行处理之前，添加了一层**FPN特征融合结构**，使底层特征图融合高层的语义信息，提高模型**对小目标的检测能力**。



改进结构后模型mAP从86.7提升至89.9



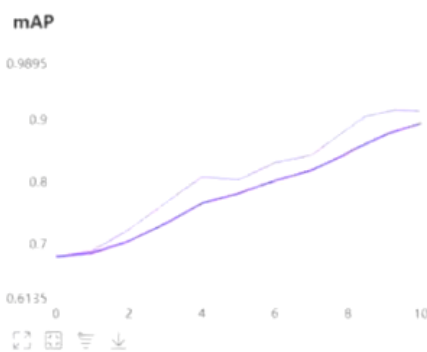
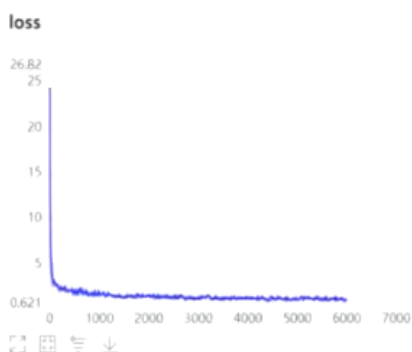
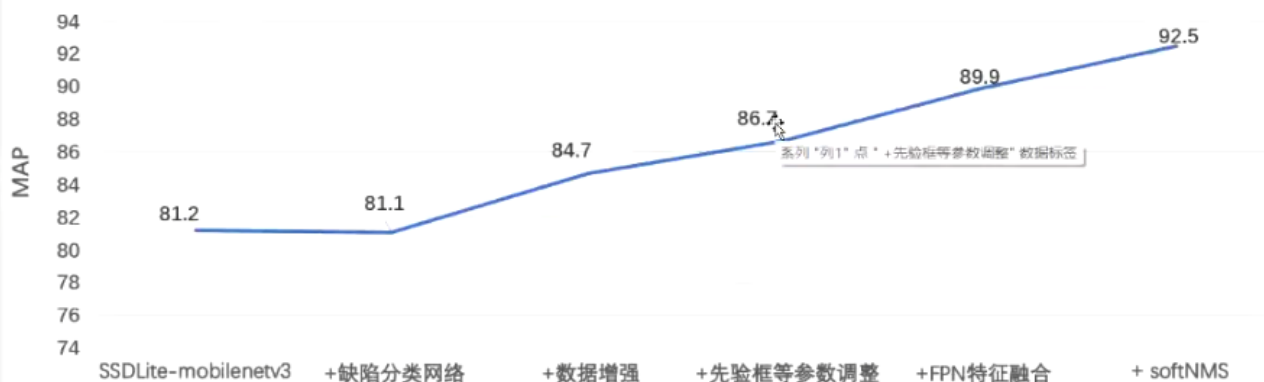
添加FPN 特征融合结构



- 结果对比

优化对比

模型测试结果



检测速度

9.87 ms/张

目标检测评价指标mAP

92.50

改进前和改进后的对比。

0x03 - A03 网络零售平台异常商品识别

1. 数据分析赛题的处理方式

- 开发工具: Python
- 使用的库: sklearn(机器学习算法库), pandas(数据处理), numpy
- 一般流程

1. 数据清洗 - 对空缺数据的处理 (如销量没有)
2. 特征工程 - 抽取需要的特征, 降维操作
3. 数据处理
 - 机器学习: 分类、聚类
 - 神经网络与深度学习
4. 前端展示

算法题, 重点在于**评价指标** (评判算法好不好, 如精确率、召回率等), 前端只是必要而不重要。

2. 时间安排

- 12月 - 准备期末考
- 1月 - 开始开发，学习相关知识
- 2月 - 继续开发，实现所有企业方要求的功能（除了前端）
- 3月 - 模型微调，起草文档
- 4月 - 最后的美化，提交所有材料
- 5月 - 区域赛答辩
- 6~7月 - 进一步优化
- 8月 - 最终答辩

3. 建议

- 前期确定方案时，最好看**学位论文**。
算法题创新较难，因此学习别人的算法。
- 做视频若外包，找靠谱的做。
- 假期不摆，答辩多练。

【后两位是设计类或B类题目，就不怎么记了.....