

全国大学生智能汽车竞赛

讯飞-智慧农业挑战赛

技

术

报

告

参赛学校 西交利物浦大学

队伍名称 GMaster

领队老师 李华康

指导老师 李华康

参赛学生 张昱轩，顾羚旦，梁森韦，任晋昊，王亦方

二零二三年

关于技术报告使用授权的说明

本人完全了解 2023 全国大学生智能汽车竞赛创意组竞赛关于保留、使用技术报告的规定，即参赛作品著作权归参赛者本人所有，比赛组委会和科大讯飞（苏州）科技有限公司可以在相关主页上收录并公开参赛作品的技术报告以及参赛作品的视频、图像资料，并将相关内容进行编纂收录。

陈乾旦 任晋昊

参赛队员签名：张星轩 梁森韦

王方

指导老师签名：

李华康

日期：2023 年 7 月 19 日

目 录

第 1 章 项目背景及意义.....

第 2 章 项目方案设计.....

第 3 章 项目算法功能.....

第 4 章 项目实施过程.....

第 5 章 项目数据分析.....

第 6 章 项目作品总结.....

参考文献

附录

第 1 章 项目背景及意义

本项目为 2023 年全国大学生智能汽车竞赛讯飞-智慧农业赛道竞赛。本项目主要解决智能小车在固定的场地内，在包含平地和斜坡的赛道上。完成视觉检测，避障，语音播报的任务。同时，应该保证在一定的时间内完成上述任务。

该项目使用了卷积神经网络，尤其是轻量化的图像分类网络 **MobilenetV2**，在有限的硬件资源上实现了高效的农作物检测，包括水果和植被的精准检测，数量统计功能。同时，队伍设计了部分鲁棒性优化操作，使得其识别效果远超于往年的传统视觉方案。能够胜任今年赛事更复杂的分类和计数任务。

该项目优化了智能小车的识别和运动逻辑，保证了智能小车能在运动的过程中识别而无需停车，缩短了任务完成的时间。该项目还对车辆的运动算法进行了调优，人工优化了小车在部分节点的运动轨迹，进一步缩短任务完成时间。

该项目的完成，将使得学生对无人驾驶，导航，视觉识别，神经网络等网络具有更深刻的知识积累和实践经验。能使得学生更加深度的了解无人驾驶和自动机器人在农业生产活动中的应用，推进科技在应用场景落地，为未来 **AI** 和机器人的应用领域做好充分的准备。

第 2 章 项目方案设计

本方案的设计可以分为四个部分：

1. 整体设计

算法的整体设计由以下几个部分构成，使智能小车能顺利完成恩任务：

1. 语音唤醒小车，启动导航模式；
2. 使用参数服务器，设定具体坐标点，使小车实现定点导航；
3. 用 **action** 方式判断小车是否成功到达目标点；
4. 到达识别点后，使用服务通讯方式向视觉发送请求，视觉返回作物类型；
5. 在小车成功到达终点之后，使用语音播报通知用户。

2. 路径规划和导航设计

运动规划导航部分是 **ROS** 自主移动小车的核心部分，主要分为全局规划和局部规划两个部分。

全局规划采用了 **A***算法，该算法可以在二维图中搜索最优路径，具有高效性和可靠性。局部规划采用了 **teb** 算法，该算法可以在不同的环境下生成平滑优化的轨迹，并且能够根据动态障碍物实时调整车辆的路径。

导航功能包集 **navigation** 中提供了 **move_base** 功能包，可以方便地实现运动规划导航功能。**move_base** 功能包提供了基于动作(action)的路径规划实现，**move_base** 可以根据给定的目标点，控制机器人底盘运动至目标位置，并且在运动过程中会连续反馈机器人自身的姿态与目标点的状态信息。

3. 视觉识别设计

视觉设计采用了神经网络替代了往届的模板匹配传统视觉算法。同时，我们最终没有使用一步到位的目标检测算法，而是使用了较为稳定的，保守，易操作的图像分类算法，通过手动设计 ROI，避免对全图进行扫描，减少不必要的运算来提升模型的运行效率。同时，我们选择将神经网络部署在图形计算卡（GPU）

4. 参数优化设计

团队中的一名队员专门进行 TEB 算法的调参，与往年使用的 DWA 算法不同，TEB 算法的引入通过弹性时间带使小车在保证安全的情况下以最短路径和时间到达目标点。为了节省算力，我们也没有开启全部的 TEB 控制，而是舍弃了一部分收效甚微的计算命令，从而提高计算效率，减少不必要的计算时间。

第 3 章 项目算法功能

1. 计算机视觉

视觉识别板块由一个 MobileNetV2^[1]图像分类网络 and 传统视觉组合完成。队员使用车辆在模拟的场地内录制数据集，用于训练神经网络，训练图像大小为 224x224。

在推理中，虽然摄像机的分辨率是 640x480。但是，由于识别版的几何特征和识别距离，我们并不需要对全图进行检测，仅需要对图片的中间部分进行检测，因此，图像坐标系中的 Y 轴将会很多部分是无用的。因此，我们团队通过认为设定的三个 ROI，每个 ROI 的大小均为 270x240，用于扫描距离 50cm 的识别板，经过实际测试，使用连续三个有重叠的 ROI 并进行扫描，即可完成全部的识别。

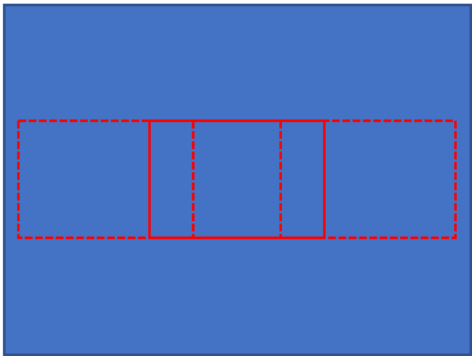


Figure 1 手动切割的 ROI，以实现最低成本的识别

在分别处理三个人工设定的目标框后，我们对三个目标框按照以下要求逻辑进行操作：

1. 如果三张图片内出现了两个相同的标签，则忽略其中一个
2. 三张图片内，在植被中出现了两种不同的标签，选择置信度高的标签，并保留置信度低的标签，是否与之后的识别冲突，如果冲突，则判定两个冲突标签，选择置信度高的标签。
3. 每个标签都必须在连续三帧内均为同一标签，才能被视为是同一个作物。

这里是一个例子：

车辆在进入到第一个房间时，检测到了三个 ROI 图片分别为 1:玉米植被，2: 玉米植被，3: 黄瓜植被。但是只有 2. 玉米植被 49% 3: 黄瓜植被 71% 持续了 3 帧。此时，车辆保留这两个选项，到第二个房间 结果为 1. 玉米植被 2. 玉米植被 3.小麦植被 三帧之后保留的为 2. 玉米植被 80%。则第二个房间被确定为玉米植被。

2. 建图，导航和避障

在建图过程中我们使用了 `cartographer` 和 `gmapping` 两种算法，但是由于机器人的轨迹或传感器的数据可能会存在偏移，导致建图不准确。这可能是由于机器人的运动不稳定、传感器的噪声、环境的复杂性等因素导致的。所以我们根据比赛地图绘制了一个标准图。

在考虑全局规划算法时，我们曾考虑过 D^* 算法。 D^* 算法是一种增量式路径规划算法，能够在不断变化的环境中动态更新路径。但是在实际测试中，我们发现 D^* 算法存在一些问题。首先， D^* 算法需要在运行时不断更新路径，这会导致计算量增大，从而影响实时性。其次， D^* 算法在处理较为复杂的环境时，存在路径不稳定、计算不准确等问题。因此，我们最终选择了 A^* 算法作为全局规划策略， A^* 算法能够在保证计算速度的前提下，得到较为准确的路径规划结果。

在局部规划算法选择中，我们一开始使用 `DWA`，但是在实际测试中发现，在处理复杂障碍物场景时，`DWA` 算法的规划结果并不理想，路径不够平滑，容易产生抖动，且调参难度较大。因此，我们最终选择了 `TEB` 算法作为局部规划策略，它能够更好地考虑局部环境信息，生成更加平滑、自然、可靠的轨迹，从而提高了小车能够更好避障。

3. 动态调参

1. 初始化阶段，通过添加关于动力学和运动学约束的默认时间信息，将初始路径变为初始轨迹。
2. 计算每个状态点之间的时间间隔，并将其作为时间弹性带。
3. 对于每个时间间隔，计算出一个窗口，该窗口包含所有可能的状态点。
4. 对于每个窗口，计算出每个状态点的代价函数值。
5. 选择代价函数值最小的状态点作为局部最优解，并将其加入到下一个时间间隔中。
6. 重复步骤 3-5,直到达到预设的迭代次数或满足其他终止条件。

第 4 章 项目实施过程

1. 视觉部分

数据采集和标注

我们团队使用自己的智能小车，对校园内 1:1 搭建的模拟真实赛道进行实车操控，获取了大致 5 小时的相机内陆数据。该项工作进行时，团队还没有确定使用目标检测算法还是采用图像分类算法进行训练。因此，队伍使用了 `YOLO` 格式进行标注。最终数据标注为训练集 5000 张，验证集 1000 张，测试数据为现场跑车调试，未设置测试集。

模型训练

在模型选择中，我们团队曾想部署 `YOLOv3` 网络作为识别，但是遇到了以下困难：

1. YOLOv3 的 Pytorch 环境训练导出 ONNX 文件无法成功转换为 TensorRT 部署上车，经过分析，认为是 TensorRT 版本过低，调试难度过高。
2. YOLOv3 的 darknet 训练环境导出的 onnx 能够成功转换为 TensorRT 并部署上车，但是在实际运行中，出现了分类不准，瞄框不准，以及推理帧率太低的三大致命劣势。经过多日的调试，依旧无法成功。此外，我们团队的显卡和 cuda 版本并不适合配置 darknet。

综上所述，我们团队放弃了使用 YOLOv3 作为部署，转而使用更容易部署的图像分类模型。我们团队查询了智能车配置的 TensorRT 版本支持的最先进的分类模型列表，并最终决定采用 MobileNetV2 模型。以下使我们选择的理由：

1. MobileNetV2 模型的速度快，网络小，训练难度低，在团队已有的 Pytorch 1.2.0 镜像中即可正常训练。
2. MobileNetV2 导出的 ONNX 支持直接转换为 TensorRT 文件并使用 cuda 进行加速。
3. 在实际赛场中，可以通过人工设定预选框使得图像分类达到类似目标检测的效果。

我们团队使用 Pytorch 架构进行神经网络训练，并使用 Jetson-inference 库进行模型部署。最终，实现了 28 帧的识别速度。

2. 调参部分

在小车导航的算法选择中，我们首先使用了往届的 DWA 算法，但是小车仅仅在局部的路径规划方面表现良好，着眼全局小车的运动轨迹并没有达到预期的效果。对于动态障碍物，小车很难进行安全且有效率的前进。

因此，我们选择了更为先进的 TEB 算法，而 TEB 算法拥有以下优点：

1. 适用于各种常见车模，如差速、全向、阿克曼模型。
 2. 有很强的前瞻性，对前方一段轨迹进行优化。
 3. 动态避障，对动态障碍物有良好的避障效果，可直接使用其封装好的障碍类 `obstacle`。
- 在使用 TEB 算法后，小车的运行和避障效率得到了显著提升。

3. 语音部分

在语音播报部分，选择使用使用 ROS 的 `sound_play` 包来实现机器人的音频播放功能。`sound_play` 包是 ROS 提供的一个用于播放音频的包。它可以播放音频文件，生成语音，播放内置的声音，等等。在比赛中，我们使用它来播放指定的音频文件。

由于播放的音频较长，实时生成语音并不现实，所以我们选择根据视觉的识别结果将准备好的音频进行组合播放。音频文件格式为 WAV，其它格式，如 MP3 需要进行解码，故舍弃。

在代码实现中，首先，我们需要创建一个 `sound_play::SoundRequest` 消息。这个消息包含了一些字段，如下：

sound：指定要播放的声音类型。在我们的例子中，我们设置为 `sound_play::SoundRequest::PLAY_FILE`，表示播放文件。

arg：指定音频文件的路径。

arg2：这个字段在播放文件时不用。

command：指定命令类型。在我们的例子中，我们设置为

`sound_play::SoundRequest::PLAY_ONCE`，表示只播放一次。

`volume`：指定音量大小。

然后，我们需要创建一个发布者，用来发布这个消息。在我们的例子中，发布者的名字是 `"sound_pub"`。

最后，我们调用发布者的 `publish()` 方法，发布消息。这将触发 `sound_play` 节点播放指定的音频文件。

第 5 章 项目数据分析

1. 视觉部分

在验证集中，模型的表现能力如下表格所示

mAP@0.5	mAP@0.95	FPS	参数文件大小
0.845	0.505	28.1	5.2M

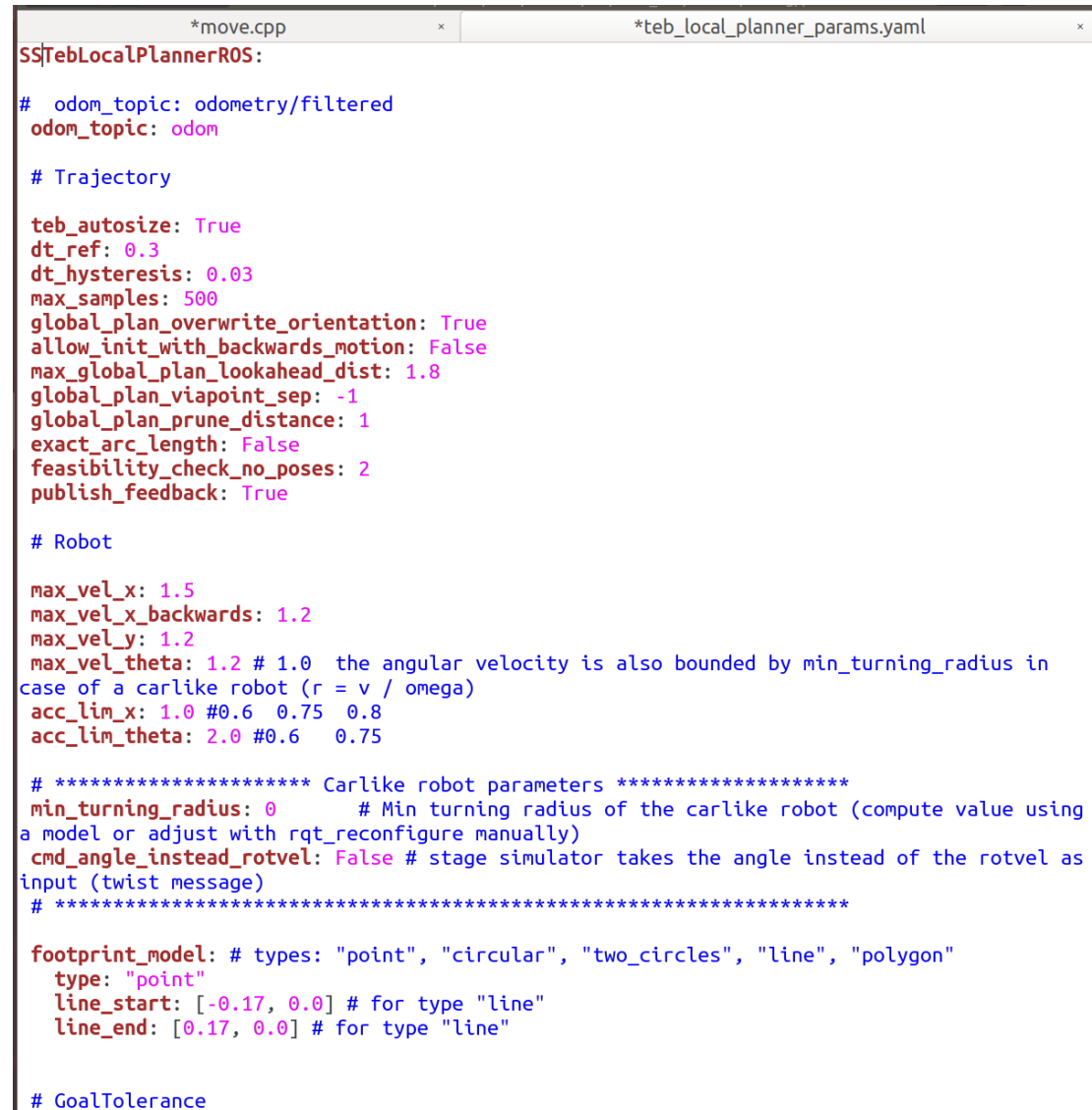
经过车辆测试，在实际运行中，视觉模型能够顺利的跟上车辆的移动节奏，并且返回正确的数据。

2. 语音部分

之前的音频文库为 MP3 格式一直无法播放，后换用 WAV 格式，并用 ROS 自带的 `sound_play` 节点进行播放。

3. 参数部分

下图为最后我们团队设定的部分参数。在调试使用 TEB 算法后，小车的运行和避障效率得到了显著提升。



```
SS|TebLocalPlannerROS:

# odom_topic: odometry/filtered
odom_topic: odom

# Trajectory

teb_autosize: True
dt_ref: 0.3
dt_hysteresis: 0.03
max_samples: 500
global_plan_overwrite_orientation: True
allow_init_with_backwards_motion: False
max_global_plan_lookahead_dist: 1.8
global_plan_viapoint_sep: -1
global_plan_prune_distance: 1
exact_arc_length: False
feasibility_check_no_poses: 2
publish_feedback: True

# Robot

max_vel_x: 1.5
max_vel_x_backwards: 1.2
max_vel_y: 1.2
max_vel_theta: 1.2 # 1.0 the angular velocity is also bounded by min_turning_radius in
case of a carlike robot (r = v / omega)
acc_lim_x: 1.0 #0.6 0.75 0.8
acc_lim_theta: 2.0 #0.6 0.75

# ***** Carlike robot parameters *****
min_turning_radius: 0 # Min turning radius of the carlike robot (compute value using
a model or adjust with rqt_reconfigure manually)
cmd_angle_instead_rotvel: False # stage simulator takes the angle instead of the rotvel as
input (twist message)
# *****

footprint_model: # types: "point", "circular", "two_circles", "line", "polygon"
  type: "point"
  line_start: [-0.17, 0.0] # for type "line"
  line_end: [0.17, 0.0] # for type "line"

# GoalTolerance
```

第 6 章 项目作品总结

该项目经过两个月的精心筹备和执行，已圆满达成任务目标，并在预定时间内顺利完成。我们的团队运用了一系列先进且鲁棒性强的算法，确保了项目的顺利进行。

从项目的立项到参赛的全过程，每一位队员都付出了巨大的努力和汗水。项目中涉及的环节繁多，包括数据录制、标注、TEB 算法参数调整、赛道搭建等等。这些看似枯燥的环节，其实锻炼了队员的耐心和毅力，使我们能够沉着应对大项目的挑战。

在项目的实施过程中，队员们不仅学习到了许多前沿的实用算法，为他们未来的学习和研究之路拓宽了视野，同时，项目的实施也增进了队员们之间的友谊，弘扬和培养了我们的团队精神。

参考文献

[1]Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Presented at the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT. <https://doi.org/10.1109/cvpr.2018.00474>

[2] Steven, M(2006). *Planning Algorithms*.

Available at: <https://www.cambridge.org/core/books/planning-algorithms/FC9CC7E67E851E40E3E45D6FE328B768>

[3]Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., ... & Ng, A. Y. (2009, May). ROS: an open-source Robot Operating System. In ICRA workshop on open source software (Vol. 3, No. 3.2, p. 5).