



(12) 发明专利

(10) 授权公告号 CN 111292353 B

(45) 授权公告日 2023. 12. 19

(21) 申请号 202010068640.1

(22) 申请日 2020.01.21

(65) 同一申请的已公布的文献号
申请公布号 CN 111292353 A

(43) 申请公布日 2020.06.16

(73) 专利权人 成都恒创新星科技有限公司
地址 610207 四川省成都市高新区天府四
街199号A座1402号

(72) 发明人 丁元一 王铭宇 喻韵旋 吴晨

(74) 专利代理机构 成都弘毅天承知识产权代理
有限公司 51230
专利代理师 杨保刚

(51) Int. Cl.

G06T 7/246 (2017.01)

G06T 7/00 (2017.01)

G06V 10/764 (2022.01)

G06V 10/82 (2022.01)

(56) 对比文件

CN 106935038 A, 2017.07.07

CN 109784306 A, 2019.05.21

CN 109817013 A, 2019.05.28

DE 102016123887 A1, 2017.06.22

US 2017249840 A1, 2017.08.31

WO 03029046 A1, 2003.04.10

CN 110163107 A, 2019.08.23

CN 109934848 A, 2019.06.25

CN 109087510 A, 2018.12.25

CN 106250838 A, 2016.12.21

JP 2013096092 A, 2013.05.20

US 2019130583 A1, 2019.05.02

US 2017323440 A1, 2017.11.09

CN 107742306 A, 2018.02.27

US 2016275369 A1, 2016.09.22

KR 101987618 B1, 2019.06.10

DE 102011087797 A1, 2013.06.06

IN 201921051720 A, 2020.01.03

GB 201913111 D0, 2019.10.23

US 2014214276 A1, 2014.07.31

WO 2012125687 A2, 2012.09.20

CN 101236656 A, 2008.08.06

CN 101662583 A, 2010.03.03

CN 102184550 A, 2011.09.14

(续)

审查员 赵萃菊

权利要求书3页 说明书8页 附图1页

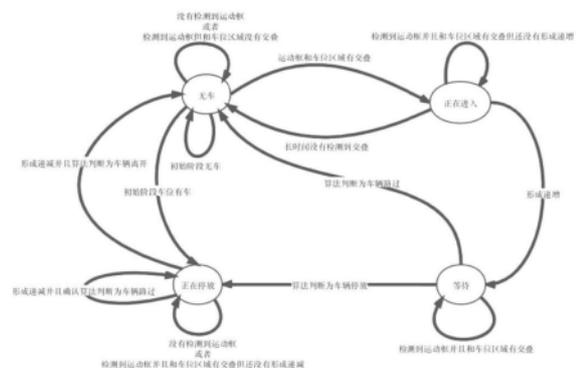
(54) 发明名称

一种停车状态变化识别方法

(57) 摘要

本发明公开了一种停车状态变化识别方法,属于停车状态识别技术领域;首先利用yolov3深度学习模型获得初始图像中的车位框,利用所述车位框判断车位的初始状态;再对后续图像中的每一帧图像,利用帧差法运动检测判断是否有运动物体;接着判断运动框与车位框是否重叠,若重叠,则利用yolov3深度学习模型判断该运动物体是否为车辆,若为车辆,则利用yolov3深度学习模型获得该运动物体的真实轮廓,利用所述真实轮廓与车位框的重合度判断车位状态;本发明降低了对深度学习模型的依赖,使得整个算法更加的低功耗,高效率,否则忽略该帧图像;若不重

叠,则车位状态保持不变;本发明降低了对深度学习模型的依赖,使得整个算法更加的低功耗,高效率。



CN 111292353 B

[接上页]

(56) 对比文件

CN 102496276 A, 2012.06.13

谢剑斌等. 2. 多帧差分法. 《视觉感知与智能视频监控》. 国防科技大学出版社, 2012,

熊俊; 陈临强. 基于多摄像头协同模式的智

能停车场管理系统. 机电工程. 2010, (第04期), 全文.

Andrew P. Brna et al.. Uncertainty-based modulation for lifelong learning. 《Neural Networks》. 2019,

1. 一种停车状态变化识别方法,其特征在于:包括以下步骤:

步骤1:利用yolov3深度学习模型获得初始图像中的车位框,利用所述车位框判断车位的初始状态;

步骤2:对后续图像中的每一帧图像,利用帧差法运动检测判断是否有运动物体,若有,产生运动框并跳转至步骤3,否则重复步骤2;

步骤3:判断运动框与车位框是否重叠,若重叠,则利用yolov3深度学习模型判断该运动物体是否为车辆,若为车辆,则跳转至步骤4,否则忽略该帧图像;若不重叠,则车位状态保持不变;

步骤4:利用yolov3深度学习模型获得该运动物体的真实轮廓,利用所述真实轮廓与车位框的重合度判断车位状态;

步骤3中,具体筛选方法为:

首先将画面一分为二,左边是马路,右边是车位,首先直接筛除坐标在左边的车辆,对于右边的车辆,只筛除两类车辆,一类是无关车辆,另一类是与停车位有关并且确认已经停稳的车辆;

在判断车辆真实轮廓坐标到车位框距离时,采用透视变换将车位梯形恢复成矩形并算出真实轮廓的新坐标;真实轮廓的坐标值采用四个数值表示:最左,最上,最右,最下;

当一号车位的状态为empty,二号车位的状态也为empty的时候,将真实轮廓的“最下”坐标满足在画面位置上高于,在数值上小于一号车位顶边的车辆以及那些真实轮廓的“最上”坐标满足在画面位置上低于,在数值上高于二号车位底边的车辆筛除;

当一号车位的状态为waiting,二号车位的状态也为waiting的时候,将真实轮廓的“最下”坐标满足在画面位置上高于,在数值上小于一号车位顶边的车辆以及那些真实轮廓的“最上”坐标满足在画面位置上低于,在数值上高于二号车位底边的车辆筛除;

当一号车位的状态为parking,二号车位的状态为empty的时候;当一号车位的状态为parking,二号车位的状态为waiting的时候;当一号车位的状态为empty,二号车位的状态为parking的时候;当一号车位的状态为waiting,二号车位的状态为parking的时候;以上情况下如果考虑的是处于parking状态的停车位,都是将那些真实轮廓的“最下”坐标满足在画面位置上高于,在数值上小于一号车位顶边的车辆以及那些真实轮廓的“最上”坐标满足在画面位置上低于,在数值上高于二号车位底边的车辆筛除;

当一号车位的状态为parking,二号车位的状态为empty的时候;当一号车位的状态为parking,二号车位的状态为waiting的时候;当一号车位的状态为empty,二号车位的状态为parking的时候;当一号车位的状态为waiting,二号车位的状态为parking的时候;以上情况下如果考虑的是处于非parking状态的停车位,都是将那些真实轮廓的“最下”坐标满足在画面位置上高于,在数值上小于当前车位顶边的车辆以及那些真实轮廓的“最上”坐标满足在画面位置上低于,在数值上高于当前车位底边的车辆筛除;

所述步骤4中,采用状态机判断车位状态,状态机包括四种状态:无车、正在进入、等待和正在停放。

2. 根据权利要求1所述的一种停车状态变化识别方法,其特征在于:所述步骤1中车位的初始状态包括无车或者正在停放。

3. 根据权利要求1所述的一种停车状态变化识别方法,其特征在于:所述步骤2中,所述

帧差法运动检测包括以下步骤:

步骤2.1 保存当前帧图像的前两帧历史图像;

步骤2.2将当前帧图像与上一帧历史图像做对比,计算两帧间的区别,若区别小于预设阈值,则判断出当前帧图像中不存在运动物体,不产生运动框,并跳转至步骤2.3;若区别不小于预设阈值,则判断当前帧图像中具有运动物体,产生运动框,跳转至步骤3;

步骤2.3 将当前帧图像与另外一帧历史图像做对比,计算两帧间的区别,若区别小于预设阈值,则判断出当前帧图像中不存在运动物体,不产生运动框,重复步骤2;若区别不小于预设阈值,则判断当前帧图像中具有运动物体,产生运动框,跳转至步骤3。

4. 根据权利要求1所述的一种停车状态变化识别方法,其特征在于:无车阶段分析步骤为:判断当前帧图像的真实轮廓与车位框的重叠度是否高于重叠度阈值,若高于重叠度阈值,则将该帧图像的重叠度、运动框的坐标、真实轮廓的坐标存入coverage_queue,下一帧状态机进入正在进入阶段进行分析;若不高于重叠度阈值,则下一帧状态机停留在无车阶段进行分析。

5. 根据权利要求4所述的一种停车状态变化识别方法,其特征在于:正在进入阶段分析步骤为:

步骤4.2.1 计算时间T内图像的重叠度,判断是否具有重叠度高于重叠度阈值的图像,若有,则将该帧图像的重叠度数值、运动框的坐标、真实轮廓的坐标存入coverage_queue,并跳转至步骤4.2.2;若没有,否则清空coverage_queue,下一帧状态机返回无车阶段进行分析;

步骤4.2.2 判断coverage_queue中是否具有三个呈递增状态的重叠度,若没有,则下一帧状态机保留在正在进入阶段进行分析;若有,下一帧状态机进入等待阶段进行分析。

6. 根据权利要求5所述的一种停车状态变化识别方法,其特征在于:所述等待阶段分析步骤如下:

步骤4.3.1 判断当前帧图像的真实轮廓与车位框的重叠度是否高于重叠度阈值,若高于重叠度阈值,将该帧图像的重叠度、运动框的坐标、真实轮廓的坐标存入coverage_queue;否则跳转至步骤4.3.2;

步骤4.3.2 计算时间T内图像的重叠度,判断是否具有重叠度高于重叠度阈值的图像,若有,则重新计时;若没有,跳转至步骤4.3.3;

步骤4.3.3 利用coverage_queue存入的真实轮廓的坐标值,判断车辆是停车还是路过,若为路过,则清空coverage_queue,下一帧状态机进入无车阶段进行分析;若为停车,则下一帧状态机进入正在停放阶段进行分析。

7. 根据权利要求6所述的一种停车状态变化识别方法,其特征在于:所述正在停放阶段分析步骤如下:

步骤4.4.1判断当前帧图像的真实轮廓与车位框的重叠度是否高于重叠度阈值,若高于重叠度阈值,将该帧图像的重叠度、运动框的坐标、真实轮廓的坐标存入coverage_queue,并跳转至步骤4.4.2,否则下一帧状态机停留在正在停放阶段进行分析;

步骤4.4.2 计算时间T内图像的重叠度,判断是否具有重叠度高于重叠度阈值的图像,若没有,则清空coverage_queue,下一帧状态机停留在正在停放阶段进行分析;若有,则跳转至步骤4.4.3;

步骤4.4.3;判断coverage_queue中是否具有三个呈递减状态的重叠度,若没有,下一帧状态机停留在正在停放阶段进行分析;若有,则判断车辆是路过还是离开,若为路过,则清空coverage_queue,下一帧状态机停留在正在停放阶段进行分析;若为离开,则下一帧状态机进入无车阶段进行分析。

一种停车状态变化识别方法

技术领域

[0001] 本发明涉及停车状态识别技术领域,具体涉及一种停车状态变化识别方法。

背景技术

[0002] 随着车辆的增长,城市街道路边往往设置有很多停车位,目前停车收费主要采用人工方式,一段街道就需要一名收费员,光收费即需要耗费大量的人力资源。而目前城市街道路停车自动识别还不是很成熟,主要有以下两类:第一类是用yolo深度学习模型追踪视频中出现的车辆,当这辆车的在视频中的坐标和提前设置好的车位坐标在一定程度上重合之后,判断为车辆进入,同理判断车辆离开,采用该类方法,视频图像中的每帧图像均需要输入yolo深度学习模型进行识别,功耗特别高;第二类是依赖一些传统的图像处理算法,这类方法缺乏鲁棒性,不能解决现实场景中的很多复杂情况,街道行人走动,灯光变化,阳光变化,白天黑夜等等。

发明内容

[0003] 本发明的目的在于:为了解决上述问题,本发明提供了一种停车状态变化识别方法。

[0004] 本发明采用的技术方案如下:

[0005] 一种停车状态变化识别方法,包括以下步骤:

[0006] 步骤1:利用yolov3深度学习模型获得初始图像中的车位框,利用所述车位框判断车位的初始状态;

[0007] 步骤2:对后续图像中的每一帧图像,利用帧差法运动检测判断是否有运动物体,若有,产生运动框并跳转至步骤3,否则重复步骤2;

[0008] 步骤3:判断运动框与车位框是否重叠,若重叠,则利用yolov3深度学习模型判断该运动物体是否为车辆,若为车辆,则跳转至步骤4,否则忽略该帧图像;若不重叠,则车位状态保持不变;

[0009] 步骤4:利用yolov3深度学习模型获得该运动物体的真实轮廓,利用所述真实轮廓与车位框的重合度判断车位状态。

[0010] 进一步的,所述步骤1中车位的初始状态包括无车或者正在停放。

[0011] 进一步的,所述步骤2中,所述帧差法运动检测包括以下步骤:

[0012] 步骤2.1 保存当前帧图像的前两帧历史图像;

[0013] 步骤2.2将当前帧图像与上一帧历史图像做对比,计算两帧间的区别,若区别小于预设阈值,则判断出当前帧图像中不存在运动物体,不产生运动框,并跳转至步骤2.3;若区别不小于预设阈值,则判断当前帧图像中具有运动物体,产生运动框,跳转至步骤3;

[0014] 步骤2.3 将当前帧图像与另外一帧历史图像做对比,计算两帧间的区别,若区别小于预设阈值,则判断出当前帧图像中不存在运动物体,不产生运动框,重复步骤2;若区别不小于预设阈值,则判断当前帧图像中具有运动物体,产生运动框,跳转至步骤3。

[0015] 进一步的,所述步骤4中,采用状态机判断车位状态,状态机包括四种状态:无车、正在进入、等待和正在停放。

[0016] 进一步的,无车阶段分析步骤为:判断当前帧图像的真实轮廓与车位框的重叠度是否高于重叠度阈值,若高于重叠度阈值,则将该帧图像的重叠度、运动框的坐标、真实轮廓的坐标存入coverage_queue,下一帧状态机进入正在进入阶段进行分析;若不高于重叠度阈值,则下一帧状态机停留在无车阶段进行分析。

[0017] 进一步的,正在进入阶段分析步骤为:

[0018] 步骤4.2.1 计算时间T内图像的重叠度,判断是否具有重叠度高于重叠度阈值的图像,若有,则将该帧图像的重叠度数值、运动框的坐标、真实轮廓的坐标存入coverage_queue,并跳转至步骤4.2.2;若没有,否则清空coverage_queue,下一帧状态机返回无车阶段进行分析;

[0019] 步骤4.2.2 判断coverage_queue中是否具有三个呈递增状态的重叠度,若没有,则下一帧状态机保留在正在进入阶段进行分析;若有,下一帧状态机进入等待阶段进行分析。

[0020] 进一步的,所述等待阶段分析步骤如下:

[0021] 步骤4.3.1 判断当前帧图像的真实轮廓与车位框的重叠度是否高于重叠度阈值,若高于重叠度阈值,将该帧图像的重叠度、运动框的坐标、真实轮廓的坐标存入coverage_queue;否则跳转至步骤4.3.2;

[0022] 步骤4.3.2 计算时间T内图像的重叠度,判断是否具有重叠度高于重叠度阈值的图像,若有,则重新计时;若没有,跳转至步骤4.3.3;

[0023] 步骤4.3.3 利用coverage_queue存入的真实轮廓的坐标值,判断车辆是停车还是路过,若为路过,则清空coverage_queue,下一帧状态机进入无车阶段进行分析;若为停车,则下一帧状态机进入正在停放阶段进行分析。

[0024] 进一步的,所述正在停放阶段分析步骤如下:

[0025] 步骤4.4.1判断当前帧图像的真实轮廓与车位框的重叠度是否高于重叠度阈值,若高于重叠度阈值,将该帧图像的重叠度、运动框的坐标、真实轮廓的坐标存入coverage_queue,并跳转至步骤4.4.2,否则下一帧状态机停留在正在停放阶段进行分析;

[0026] 步骤4.4.2 计算时间T内图像的重叠度,判断是否具有重叠度高于重叠度阈值的图像,若没有,则清空coverage_queue,下一帧状态机停留在正在停放阶段进行分析;若有,则跳转至步骤4.4.3;

[0027] 步骤4.4.3;判断coverage_queue中是否具有三个呈递减状态的重叠度,若没有,下一帧状态机停留在正在停放阶段进行分析;若有,则判断车辆是路过还是离开,若为路过,则清空coverage_queue,下一帧状态机停留在正在停放阶段进行分析;若为离开,则下一帧状态机进入无车阶段进行分析。

[0028] 综上所述,由于采用了上述技术方案,本发明的有益效果是:

[0029] 本发明将帧差运动检测法和yolov3进行结合,降低了对深度学习模型的依赖,使得整个算法更加的低功耗,高效率,在edge device上运行速度更快。

[0030] 本发明考虑并解决了一些实际情况中城市街道路边停车的问题,比如说由于摄像头角度(不是鸟瞰视角)所带来的拍摄到的停车位车辆以及其车牌被遮挡很严重的问题,停

车位容易有行人或者其他物体通过/逗留,有车辆利用停车位进行调头操作等情况。也避免了一天当中随着时间的变化,太阳光线照射,车位上阴影的变化所带来的影响。

附图说明

[0031] 为了更清楚地说明本发明实施例的技术方案,下面将对实施例中所需要使用的附图作简单地介绍,应当理解,以下附图仅示出了本发明的某些实施例,因此不应被看作是对范围的限定,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他相关的附图。

[0032] 图1为本发明的流程图;

[0033] 图2为本发明摄像头拍摄的示意图。

具体实施方式

[0034] 为了使本发明的目的、技术方案及优点更加清楚明白,以下结合附图及实施例,对本发明进行进一步详细说明。应当理解,此处所描述的具体实施例仅用以解释本发明,并不用于限定本发明,即所描述的实施例仅仅是本发明一部分实施例,而不是全部的实施例。通常在此处附图中描述和示出的本发明实施例的组件可以以各种不同的配置来布置和设计。

[0035] 因此,以下对在附图中提供的本发明的实施例的详细描述并非旨在限制要求保护的本发明的范围,而是仅仅表示本发明的选定实施例。基于本发明的实施例,本领域技术人员在没有做出创造性劳动的前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0036] 需要说明的是,术语“第一”和“第二”等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来,而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且,术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、方法、物品或者设备不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、物品或者设备所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括所述要素的过程、方法、物品或者设备中还存在另外的相同要素。

[0037] 以下结合实施例对本发明的特征和性能作进一步的详细描述。

[0038] 实施例

[0039] 本实施例提供一种停车状态变化识别方法,该识别方法应用于装在路灯上的摄像头上,判断受摄像头监视的车位的停车状态,算法可以支持一个摄像头监控多个停车位,本实施例让一个摄像头监控2个车位,每一个车位有一个独立的coverage_queue。

[0040] 当系统启动时,每隔一秒(可自定义)或者是每隔一(FPS*1秒)帧(可自定义),会采集图像进行运算,此刻的图片称之为当前帧,同时也会把当前帧存入长度为3的队列(queue)中。本实施例中重叠度阈值均设置为0.1。

[0041] 步骤1:系统初始阶段运行yolov3深度学习模型(车辆识别模型)对车位上是否已停有车进行判断,如果系统启动时判断车位上没有车,初始状态则为empty(无车)状态,如果系统启动时判断车位上已经有车了,初始状态则为parking(正在停放)状态。

[0042] 步骤2:利用帧差法运动检测得到后续图像中是否有运动物体,若有,产生运动框并跳转至步骤3,否则重复步骤2(每获取一个新的当前帧均要进行步骤2的判断);

[0043] 所述帧差法运动检测包括以下步骤:

[0044] 步骤2.1 为了一定程度上避免一些因为车速低(前后两帧变化不大)而出现画不出矩形框的情况发生,保存当前帧图像的前两帧历史图像,即从当前帧所在队列中获取前两帧历史图像;

[0045] 步骤2.2将当前帧图像与上一帧历史图像做对比,计算两帧间的区别,若区别小于预设阈值,则判断出当前帧图像中不存在运动物体,不产生运动框,并跳转至步骤2.3;若区别不小于预设阈值,则判断当前帧图像中具有运动物体,产生运动框,跳转至步骤3;

[0046] 步骤2.3 将当前帧图像与另一帧历史图像(前前帧图像)做对比,计算两帧间的区别,若区别小于预设阈值,则判断出当前帧图像中不存在运动物体,不产生运动框,重复步骤2;若区别不小于预设阈值,则判断当前帧图像中具有运动物体,产生运动框,跳转至步骤3。

[0047] 步骤3:判断运动框与车位框是否重叠,若重叠,则利用yolov3深度学习模型判断该运动物体是否为车辆,若为车辆,则跳转至步骤4,否则忽略该帧图像;若不重叠,则车位状态保持不变;

[0048] 由于yolov3深度学习模型会识别出整张图片中所有的物体,因此要配合运动框找到需要的那个运动物体的真实轮廓,筛除掉无关物体轮廓。帧差法运动检测所计算出来的运动框不准确,不准确体现在运动框有时候比真实轮廓大,有时比真实轮廓小。当比真实轮廓小的时候,运动框被包含于真实轮廓中,可以通过判别哪个物体的真实轮廓包含了这个运动框,从而获取真实轮廓;比真实轮廓大的时候,真实轮廓被包含于运动框中,通过哪个物体的真实轮廓被包含于这个运动框,从而获取真实轮廓,但是因为运动框过大,很容易包含其他无关车辆,比如别的停车位上的车位,马路上路过的其他车辆,所以需要进一步的筛选。

[0049] 进一步筛选方法为:首先将画面一分为二,左边是马路,右边是车位,首先直接筛除坐标在左边的车辆,对于右边的车辆,只筛除两类车辆,一类是无关车辆,比如距离太远的车辆;另一类是与停车位有关并且确认已经停稳的车辆。

[0050] 进一步筛选方法在本实施例中的具体做法需要考虑具体哪个车位以及其当前状态进行具体分析,在判断车辆真实轮廓坐标到车位框距离的时候都需要先用透视变换(Perspective Transformation)将车位梯形恢复成矩形并算出真实轮廓的新坐标。coverage_queue中真实轮廓的坐标值采用四个数值表示:最左,最上,最右,最下。最左最上定义了左上角坐标的xy值,最右最下定义了右下角坐标的xy值。

[0051] 由于本实施例中每个摄像头负责两个停车位,每个停车位都有属于自己的coverage_queue,算法会逐一考虑每个车位的停车状态。

[0052] 当一号车位的状态为empty,二号车位的状态也为empty的时候,无论考虑哪个车位的时候,都是将真实轮廓的“最下”坐标满足在画面位置上高于,在数值上小于一号车位顶边的车辆以及那些真实轮廓的“最上”坐标满足在画面位置上低于,在数值上高于二号车位底边的车辆筛除;

[0053] 当一号车位的状态为waiting,二号车位的状态也为waiting的时候,无论考虑哪个车位的时候,都是将真实轮廓的“最下”坐标满足在画面位置上高于,在数值上小于一号车位顶边的车辆以及那些真实轮廓的“最上”坐标满足在画面位置上低于,在数值上高于二

号车位底边的车辆剔除；

[0054] 当一号车位的状态为parking,二号车位的状态为empty的时候;当一号车位的状态为parking,二号车位的状态为waiting的时候;当一号车位的状态为empty,二号车位的状态为parking的时候;当一号车位的状态为waiting,二号车位的状态为parking的时候;以上情况下如果考虑的是处于parking状态的停车位,都是将那些真实轮廓的“最下”坐标满足在画面位置上高于,在数值上小于一号车位顶边的车辆以及那些真实轮廓的“最上”坐标满足在画面位置上低于,在数值上高于二号车位底边的车辆剔除;

[0055] 当一号车位的状态为parking,二号车位的状态为empty的时候;当一号车位的状态为parking,二号车位的状态为waiting的时候;当一号车位的状态为empty,二号车位的状态为parking的时候;当一号车位的状态为waiting,二号车位的状态为parking的时候;以上情况下如果考虑的是处于非parking状态的停车位,都是将那些真实轮廓的“最下”坐标满足在画面位置上高于,在数值上小于当前车位顶边的车辆以及那些真实轮廓的“最上”坐标满足在画面位置上低于,在数值上高于当前车位底边的车辆剔除。

[0056] 步骤4:利用yolov3深度学习模型获得该运动物体的真实轮廓,利用所述真实轮廓与车位框的重合度判断车位状态。

[0057] 可通过TensorRT加速yolov3深度学习模型,整个过程只需要0.6秒,小于取帧时间间隔(每隔一秒取摄像头的一帧画面),使用FPGA加速处理时间更少,运行速度会更加快。

[0058] 所述步骤4中,通过计算出来的运动框,结合状态机对造成此运动框的物体的运动意图进行判断,状态机包括四种状态:无车、正在进入、等待和正在停放;每一个当前帧都会对应一个且只有一个状态,分析当前帧的时候也是在相应的阶段进行计算和分析。例如,从结果来看假设一个停车-出车整个流程总共占用了100帧,其中前25帧是没有车,那处理这25帧的图片就按照无车阶段进行分析判断,在第25帧的时候满足了进入下一阶段的条件,当到了第26帧的时候,就会按照正在进入阶段进行分析和判断,以此类推。

[0059] 无车阶段分析步骤为:

[0060] 无车阶段分析步骤为:判断当前帧图像的真实轮廓与车位框的重叠度是否高于0.1,若高于0.1,则将该帧图像的重叠度、运动框的坐标、真实轮廓的坐标存入coverage_queue,下一帧状态机进入正在进入阶段进行分析;若不高于0.1,则下一帧状态机停留在无车阶段进行分析。

[0061] 正在进入阶段分析步骤为:

[0062] 步骤4.2.1 计算时间T内(5s)图像的重叠度,判断是否具有重叠度高于0.1的图像,若有,则将该帧图像的重叠度数值、运动框的坐标、真实轮廓的坐标存入coverage_queue,并跳转至步骤4.2.2;若没有,否则清空coverage_queue,下一帧状态机返回无车阶段进行分析;

[0063] 步骤4.2.2 判断coverage_queue中是否具有三个呈递增状态的重叠度(无需连续的三个数值,比如coverage_queue里面有5个数,第123呈递增,第135呈递增,第125呈递增等等均可),若没有,则下一帧状态机保留在正在进入阶段进行分析;若有,下一帧状态机进入等待阶段进行分析。

[0064] 所述等待阶段分析步骤如下:

[0065] 步骤4.3.1 判断当前帧图像的真实轮廓与车位框的重叠度是否高于0.1,若高于

0.1,将该帧图像的重叠度、运动框的坐标、真实轮廓的坐标存入coverage_queue;否则跳转至步骤4.3.2;

[0066] 步骤4.3.2 计算时间T内(5s)图像的重叠度,判断是否具有重叠度高于0.1的图像,若有,则重新计时;若没有,跳转至步骤4.3.3;

[0067] 步骤4.3.3 利用coverage_queue存入的真实轮廓的坐标值,判断车辆是停车还是路过,若为路过,则清空coverage_queue,下一帧状态机进入无车阶段进行分析;若为停车,则下一帧状态机进入正在停放阶段进行分析。

[0068] 其中停车和路过判断方法为:

[0069] 该方法分进车阶段和出车阶段,其中进车阶段为从waiting(等待)变成parking(正在停放),出车阶段为从parking(正在停放)变成empty(无车)。

[0070] 进车阶段,取coverage_queue中距离当前时间最远的那一次保存的真实轮廓的坐标值,这个坐标值是用四个数值表示的:最左,最上,最右,最下。最左最上定义了左上角坐标的xy值,最右最下定义了右下角坐标的xy值,这里只需要最下,也就是汽车后轮的高度来进行判断,因为车的底边是受摄像头角度,车辆本身大小的区别影响最小的参考物。每一个车位框在高度上有一个范围,如果最下的值落在了其中一个车位框的范围内,就可以判断出这辆车是停在了哪一个车位上,如图2所示,银色汽车的底边落在了前面那个车位的范围内,被判定为停在了前面那个车位。那如果最下的值不属于任何车位的范围内,则可以判断车辆没有停车,认为是路过。本实施例中每个车位的范围阈值均设置为大于负0.3小于0.9,意为车辆真实轮廓的最下坐标数值和车位框底边的垂直距离D与车位框顶边和车位框底边的垂直距离,即车位框高度L的比值。比值为正代表车辆的后轮位置处于车位区域内,比值为负意味着车辆的后轮位置超出了车位区域的底边,这一片区域也算作是车位框的范围内主要是考虑到了可能发生的停车不规范问题(压线)。

[0071] 由于摄像头具有角度,如图2所示,车位是一个梯形,需要先用透视变换(Perspective Transformation)将车位梯形恢复成矩形,再根据得到的变换矩阵(transformation matrix)算出车辆的真实轮廓坐标在矩形中的位置,也就是模拟摄像头没有角度的情况下看到的车位鸟瞰图,否则靠前车位的上底边到下底边的距离肯定是会比靠后车位的上底边到下底边的距离短的,这样对于车辆底边到各个车位的距离的判断是不准确的。

[0072] 出车阶段,取coverage_queue中距离当前时间最近的那一次保存的真实轮廓的坐标值,其余的步骤和进车阶段一致。

[0073] 所述正在停放阶段分析步骤如下:

[0074] 步骤4.4.1判断当前帧图像的真实轮廓与车位框的重叠度是否高于0.1,若高于0.1,将该帧图像的重叠度、运动框的坐标、真实轮廓的坐标存入coverage_queue,并跳转至步骤4.4.2,否则下一帧状态机停留在正在停放阶段进行分析;

[0075] 步骤4.4.2 计算时间T(5s)内图像的重叠度,判断是否具有重叠度高于重叠度阈值的图像,若没有,则清空coverage_queue,下一帧状态机停留在正在停放阶段进行分析;若有,则跳转至步骤4.4.3;

[0076] 步骤4.4.3;判断coverage_queue中是否具有三个呈递减状态的重叠度,若没有,下一帧状态机停留在正在停放阶段进行分析;若有,判断车辆是路过还是离开,若为路过,

则清空coverage_queue,下一帧状态机停留在正在停放阶段进行分析;若为离开,则下一帧状态机进入无车阶段进行分析;

[0077] 本发明中,帧差法运动检测运算量小且速度快,可以提取运动目标的轮廓,但轮廓不精准,由于算法依赖运动物体和停车位重叠程度的变化判断是否为进出场运动,这一不精确特性对算法会产生很大影响。比如帧差法画出的马路上直行驶过的一辆车的轮廓会比这辆车真实的轮廓大很多,导致会和马路边的停车位造成重叠,这样重叠度队列中会产生不真实重叠度,导致误判;因此当帧差法运动检测得到的运动框与车位框有一定的重合之后,再采用yolov3模型获得物体的真实轮廓,利用真实轮廓计算出的重叠度,不仅不用将每一帧图像均送入yolov3模型进行识别,降低了整个算法的功耗,还极大的提高了重合度的准确率。

[0078] 本发明通过帧差法运动检测,筛除掉一些相对于汽车面积更小的物体,比如树叶和行人;进一步使用yolov3模型,筛除掉所有非车物体,消除了所有非车物体对算法的影响;再将yolov3返回的车辆真实轮廓坐标与停车位的停车状态配合,筛除掉与这次运动无关的车辆。

[0079] 本发明状态机等待阶段中设置了时间阈值,筛除了短暂停车事件的影响,这样无论这辆车在车位区域做了什么,本算法都可以忽略,只判断这个车位在没有运动物体之后是处于什么状态,比如没有运动物体因为车停稳了还是车离开了。若是停稳了,再判断具体停在哪个车位上。

[0080] 本发明在进车阶段保留了进车前的多帧图像,可以用这些图像去搜索车牌,重叠度低的时刻的物体图片车牌不被遮挡的概率是很高的;在出车阶段,在当前帧形成递减并且定位算法判断是离开,为了能够避免出车过程中车牌被遮挡导致识别不出车牌的问题,启动KCF跟踪算法,将coverage_queue中最近一次的车辆真实轮廓的坐标作为KCF的初始框,跟踪这辆离开车辆的后续运动,利用后续运动的图片识别出没有被遮挡的车牌。

[0081] 由于摄像头有角度,一辆车的运动覆盖了多个车位的情况是很常见的,这就会带来一个问题,就是如果画面中停车位区域附近有两辆车同时在移动或者一前一后运动,一辆车先动,另外一辆车紧接着做出了运动,交替进行,这样一个车位的coverage_queue里面存的重叠度就不来源于同一辆车,会造成误判。解决方法就是通过车辆的真实轮廓坐标配合追踪算法进行区分,利用一辆车在相邻帧不会由于运动造成巨大位移的特性,通过真实轮廓距离的变化大小来区分出两辆相隔较远的不同的车辆;对于距离相隔较近的两辆车,则可以利用KCF追踪算法的分类器特性进行区分, KCF中的分类器可以根据轮廓,颜色等特征值将车辆进行区分。

[0082] 本发明解决了丢帧现象,若分析时间超出了一秒,则会发生丢帧现象。因此本发明提升处理每一帧图片的速度,采用tensorRT和FPGA进行模型的推断(inference)加速;创建一个buffer存储从摄像头返回的图片,按照一秒为时间间隔将摄像头的图片填充至buffer中,主程序直接从buffer中获取图片,当一张图片处理完毕后主程序会立刻从这个buffer中获取下一张图片来进行处理。假设当需要运行深度学习模型的时候处理一张图片所需要的时间是1.5秒,不需要运行的时候需要0.2秒,在需要模型的时候buffer的图片会堆积,但不需要运行的时候buffer就会被消耗,所以这个buffer不会处于一个永久增长的过程,也能有效的解决丢帧的问题。

[0083] 以上所述仅为本发明的较佳实施例而已,并不用以限制本发明,凡在本发明的精神和原则之内所作的任何修改、等同替换和改进等,均应包含在本发明的保护范围之内。

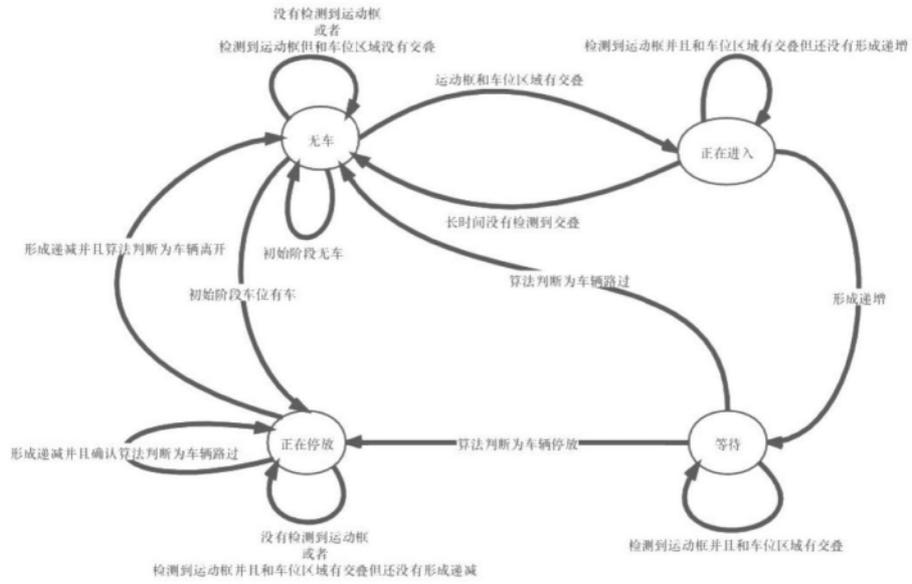


图1



图2