

行车历史数据提取模型设计与研究

张航铭

天津工业大学, 天津

收稿日期: 2023年1月16日; 录用日期: 2023年2月16日; 发布日期: 2023年2月24日

摘要

车内人机交互(HMI)系统已经发展成为汽车必备的一个“零件”。作为计算机科学、行为科学以及其他几个研究领域的交汇处, 车内HMI在实际应用领域中具有巨大的潜在价值。然而, 现有的HMI交互内容大部分为指令或者公共数据等不涉及到驾驶员和驾驶数据, 造成了HMI在个性化服务上的数据缺失问题。因此, 本文构建了行车历史数据提取模型, 通过真实环境下的行车视频数据, 提取出与行车过程中有联系的目标集合以及目标对应的时间集合和位置集合, 最后将提取出的数据持久化到本地。除此之外, 本文设计的性能测试实验证明了此模型技术方案的可行性和准确性。

关键词

车内人机交互, 行车历史数据

Design and Research of Driving History Data Extraction Model

Hangming Zhang

Tiangong University, Tianjin

Received: Jan. 16th, 2023; accepted: Feb. 16th, 2023; published: Feb. 24th, 2023

Abstract

In-vehicle Human-Machine Interaction (HMI) system has developed into an essential “part” of the car. As the intersection of computer science, behavioral science, and several other research fields, in-vehicle HMI has great potential value in the field of practical application. However, most of the existing HMI interaction content is instructions or public data, which does not involve the driver and driving data, resulting in the lack of data in the personalized service of HMI. Therefore, this paper constructs a driving history data extraction model. Through the driving video data in the real environment, the target set related to the driving process and the time set and location set

corresponding to the target are extracted, and finally the extracted data is persisted to local. In addition, the performance test experiment designed in this paper proves the feasibility and accuracy of the technical solution of this model.

Keywords

In-Vehicle Human-Computer Interaction, Driving History Data

Copyright © 2023 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

自计算机诞生以来, 人机交互系统也随之得以兴起。在现实生活中, 只要有计算机等通讯信息技术的使用, 人机交互就永远是人类不老的话题。车内人机交互系统就是人机交互的一个分支, 简单来说, 车内人机交互系统就是人与汽车通过一定的交互方式完成信息交换的过程, 它能够直接影响用户的用车体验。然而现有交互系统的交互内容大部分为指令数据或者公共数据等形式, 不涉及驾驶数据和行车数据, 因此造成了车内交互系统提供个性化服务上的数据缺失问题。

随着用户在汽车行业消费观念的转变, 除了自动驾驶、智能化座舱、新能源等服务功能, 对车内人机交互系统的需求也日益激增。国外的汽车行业最早可以追溯到 1885 年, 其人机交互系统体验和服务较为完善[1]。例如特斯拉汽车上的物理按键非常少, 一般放在方向盘上或方向盘附近。这些物理按键主要用来控制巡航、自动化驾驶、刮水器以及照明灯等功能。但是大多数辅助功能都没有专门的物理控件。奔驰汽车[2]可以为用户提供更智能化的驾驶体验, 例如: 根据驾驶者身材比例为其调整适合的后视镜角度和座椅位置、实时监测驾驶员操作动作等。虽然国产汽车起步较晚, 但是也有很多互联网企业、高校和汽车制造商进行合作开展了相关研究。到目前为止, 国产车在国民心中已经有了举足轻重的地位。蔚来汽车的 Aspen 系统搭载了 NOMI 人工智能管家, 它拥有超强的学习能力和语音指令识别等, 值得一提的是还可以在操作台上方的显示屏中显示出多种情感的表情, 就好像它是有生命的管家一样。理想 AD Max 系统以“一切为了安全”为研发的基本原则。为了应对遭遇强光或者黑暗环境时, 摄像头的能力会大打折扣, 在 BEV 视觉信息为主的基础上, 增加了激光雷达和高精地图的信息作为辅助, 独创了 BEV 融合算法。为进一步加强安全性, 还加入了视觉安全模块和激光雷达安全模块, 与 BEV 框架模型互为冗余, 能在极限环境下为安全提供多一层保障。

综上所述, 国内外对车内交互系统的个性化服务研究还处在初期阶段。因此, 本文提出了一个行车历史数据提取模型, 用来将行车过程中与行车有关联的目标集合、时间集合以及位置集合存储到本地, 为提供个性化服务提供底层数据。此外, 为了评估此模型, 本文通过行车记录仪录制了累计 28 小时的真实环境下的行车视频并设置了性能测试实验, 证实了本文所提模型的有效性和准确性。

2. 行车历史数据提取模型

2.1. 模型总体框架

如图 1 所示, 本文所提出的模型主要分为 4 个模块: GPS 数据解析模块、目标集合以及时间信息提取模块、数据整合模块以及数据持久化模块。以下将对各个模块进行详细介绍。

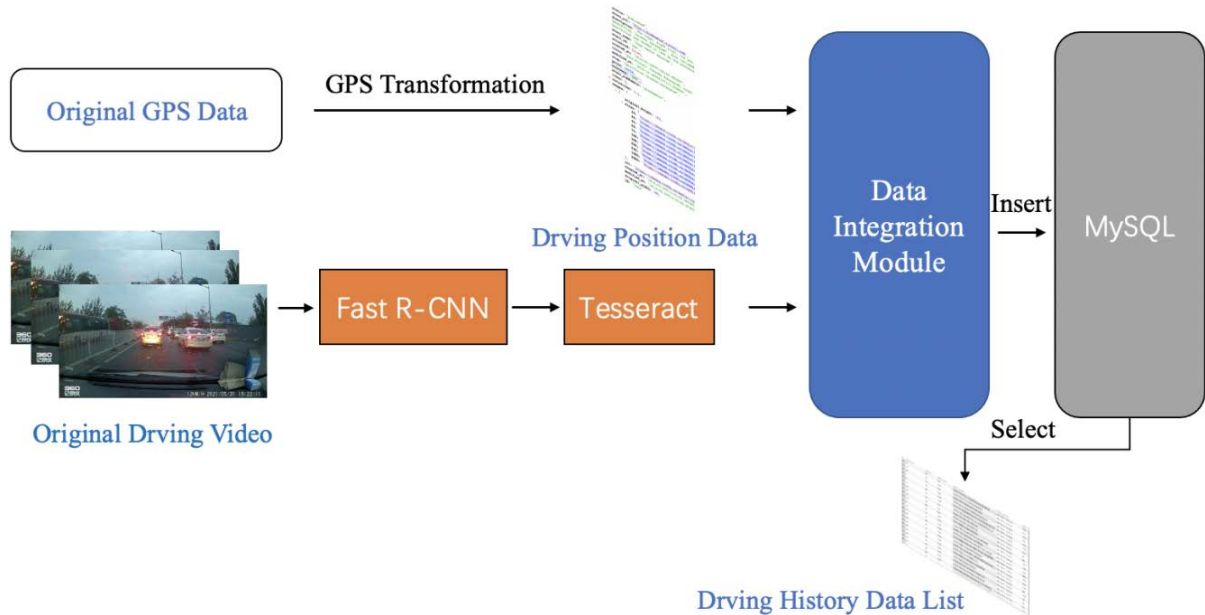


Figure 1. Model overall framework diagram
图 1. 模型总体框架图

2.2. GPS 数据解析模块

由于行车记录仪提供的位置数据信息为 GPX 格式，这是一种基于 XML 的轨道格式，也是 GPS 数据通用的一种规范格式。如图 2 所示，其中红色框里的数据表示经纬度信息，绿色框里的数据表示格林尼治时间，因此需要将二者解析为对应的地址以及对应的北京时间。在这里需要用到 gpxpy 工具，该工具是一个简单的 Python 库，用于解析和操作 GPX 格式文件。

```

+<trkpt lon="117.10843925" lat="39.06660607"></trkpt>
-<trkpt lon="117.10846063" lat="39.06671852">
  <ele>7.772674560546875</ele>
  <time>2021-05-31T11:16:21Z</time>
  <speed>3.430000066757202</speed>
</trkpt>
-<trkpt lon="117.10845101" lat="39.06690078">
  <ele>5.923065185546875</ele>
  <time>2021-05-31T11:16:25Z</time>
  <speed>5.420000076293945</speed>
</trkpt>
-<trkpt lon="117.108439" lat="39.06711999">
  <ele>3.5665283203125</ele>
  <time>2021-05-31T11:16:29Z</time>
  <speed>5.789999961853027</speed>
</trkpt>

```

Figure 2. Raw GPX format data
图 2. 原始 GPX 格式数据

首先本文使用 gpxpy 工具将原始的 GPS 数据解析为经纬度集合 $\{Lo_i, La_i\}_{i=1}^M$ 以及格林尼治时间集合 $\{Gt_i\}_{i=1}^M$ 。然后使用百度地图提供的 API 将经纬度集合转化为中国地图中实际的位置信息 $\{P_i\}_{i=1}^M$ ，另外格林尼治时间集合中的数据需要加 8 小时转为北京时间集合 $\{Bj_i\}_{i=1}^M$ ，最后将转化好的位置数据和时间数据

存储为 json 格式供后续使用。解析完后的数据如图 3 所示。

```

"2021-05-31 19:16:11": "天津市西青区精武镇西环路天津工业大学新校区",
"2021-05-31 19:16:21": "天津市西青区精武镇西环路天津工业大学新校区",
"2021-05-31 19:16:25": "天津市西青区精武镇洋缘道天津工业大学新校区",
"2021-05-31 19:16:29": "天津市西青区精武镇洋缘道天津工业大学新校区",
"2021-05-31 19:16:33": "天津市西青区精武镇西环路天津工业大学新校区",
"2021-05-31 19:16:37": "天津市西青区精武镇西环路天津工业大学新校区",
"2021-05-31 19:16:41": "天津市西青区精武镇洋缘道天津工业大学新校区",
"2021-05-31 19:16:45": "天津市西青区精武镇西环路天津工业大学新校区",
"2021-05-31 19:16:47": "天津市西青区精武镇西环路天津工业大学新校区",
"2021-05-31 19:16:51": "天津市西青区精武镇西环路天津工业大学新校区",
"2021-05-31 19:16:55": "天津市西青区精武镇西环路天津工业大学新校区",
"2021-05-31 19:16:59": "天津市西青区精武镇西环路天津工业大学新校区",
"2021-05-31 19:17:03": "天津市西青区精武镇西环路天津工业大学新校区",
"2021-05-31 19:17:07": "天津市西青区精武镇北环路天津工业大学新校区",
"2021-05-31 19:17:11": "天津市西青区精武镇北环路天津工业大学新校区",
"2021-05-31 19:17:15": "天津市西青区精武镇北环路天津工业大学新校区",
"2021-05-31 19:17:19": "天津市西青区精武镇天津市大学软件学院B区天津工业大学新校区",
"2021-05-31 19:17:23": "天津市西青区精武镇天津市大学软件学院B区天津工业大学新校区",

```

Figure 3. Parsed GPS data

图 3. 解析完成的 GPS 数据

2.3. 目标集合以及时间信息提取模块

考虑到历史数据提取任务的需求，目标集合 $\{b_i\}_{i=1}^N$ 的提取是本模型中最重要的一个部分，因此一个好的目标检测方法起着至关重要的作用。在本文中选用了 Fast R-CNN [3] 作为获取目标集合的网络模型，并只保留了置信度大于 0.8 的目标。由于轿车和行人两类目标频繁出现，因此本文剔除了这两类目标。

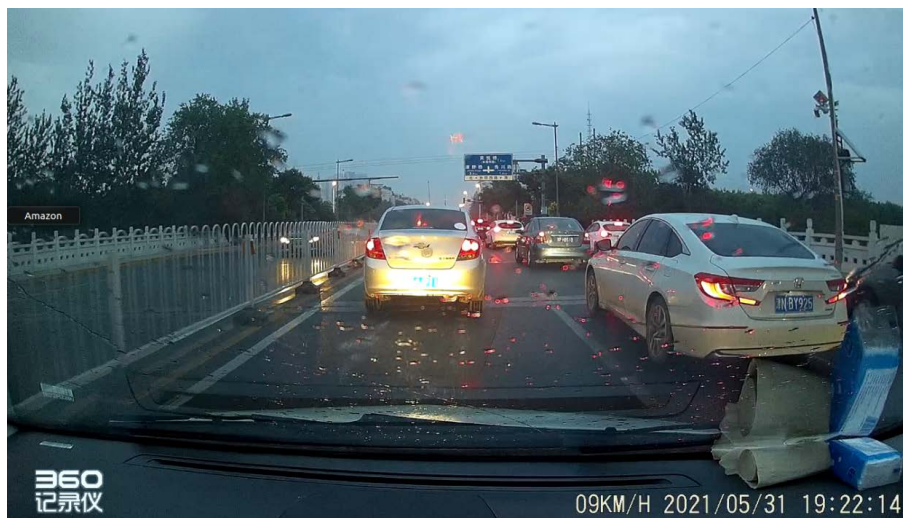


Figure 4. Example of driving image

图 4. 行车图像实例

为了更好的获得目标集合中检测到每个目标时的时间信息，本文在检测目标的同时记录了检测到该目标时的帧数信息集合 $\{f_i\}_{i=1}^N$ 。由于行车视频的右下角提供了北京时间信息，并且没有任何遮挡以及扭曲，如图 4 所示。因此本文首先将包含时间信息的图像区域裁剪下来，得到待提取时间信息的图像集合 $\{s_i\}_{i=1}^N$ 。最后使用 Tesseract [4] 工具进行图片上的光学字符识别得到检测到该目标时的时间数据集合 $\{d_i\}_{i=1}^N$ 。该工具是 Google 开发的一个基于 LSTM [5] 的 OCR 模型，具备良好的性能。

2.4. 数据整合模块

该模块负责将所有需要持久化存储的数据进行整合。首先本文将时间集合中的数据与解析出的 GPS 数据中的北京时间数据集合相匹配，得到检测到每个目标时的位置信息 $\{g_i\}_{i=1}^N$ 。时间匹配算法伪代码如图 5 所示。最终整合这些数据，得到需要持久化存储的数据集合 $\{b_i\}_{i=1}^N, \{f_i\}_{i=1}^N, \{g_i\}_{i=1}^N, \{d_i\}_{i=1}^N$ 。

```

输入:  $\{d_i\}_{i=1}^N$ : 时间集合;
        $\{Bj_i, P_i\}_{i=1}^M$ : 北京时间集合和实际位置集合;
输出:  $\{g_i\}_{i=1}^N$ : 目标位置集合;
1: 定义输出  $\{g_i\}_{i=1}^N$ ;
2: for  $I$  in  $\{d_i\}_{i=1}^N$ :
3:     定义最小匹配时间差  $C$ ;
4:     for  $J$  in  $\{Bj_i, P_i\}_{i=1}^M$ :
5:          $Time = Bj_i - d_i$ ;
6:         if  $Time > 0$  and  $Time < C$  :
7:              $g_i = P_j$ ;
8:              $C = Time$ ;
9: 输出  $\{g_i\}_{i=1}^N$ 

```

Figure 5. Pseudocode of time matching algorithm

图 5. 时间匹配算法伪代码

2.5. 数据持久化模块

该模块负责将整合好的数据持久化到 MySQL [6]数据库中，结果如图 6 所示。本文设计了两张数据库表用来分别用来存储行车视频信息以及检测出的数据集合信息。表结构如表 1，表 2 所示。

#	process_id	video_id	detect_class	detect_startGPS	detect_startTime	detect
20	17	63	bus	天津市河东区中山门街道宇顺洗车试验楼居民区	2021-05-31 19:51:10	1796
21	16	63	truck	天津市河东区中山门街道试验楼居民区	2021-05-31 19:51:00	1472
22	23	74	bus	天津市河东区向阳楼街道万东路106号	2021-05-31 19:53:50	1193
23	11	56	truck	天津市河东区向阳楼街道雍鑫·雍祥园(建设中)	2021-05-31 19:54:15	135
24	5	50	truck	天津市河东区富民路街道华昱(天津)劳保用品有限公司	2021-05-31 19:49:58	1436
25	14	62	truck	天津市河东区富民路街道天津市冶金医院综合办公室	2021-05-31 19:49:08	1738
26	12	57	bus	天津市河东区鲁山道街道紫玉园	2021-05-31 19:58:10	1797
27	38	86	bus	天津市河东区鲁山道街道紫玉园	2021-05-31 19:58:11	18
28	19	68	bus	天津市河西区友谊路街道纯真里	2021-05-31 19:44:37	804
29	30	80	truck	天津市河西区友谊路街道黑牛城道辅路联盟里	2021-05-31 19:44:10	1773
30	9	53	truck	天津市河西区天塔街道五一阳光皓日园	2021-05-31 19:41:56	1360
31	18	67	truck	天津市河西区天塔街道司法警校宿舍小区人委北里	2021-05-31 19:38:10	1790
32	44	92	truck	天津市河西区天塔街道天资里天津市地方铁路管理局	2021-05-31 19:40:11	1800
33	36	84	truck	天津市河西区天塔街道黑牛城道282号	2021-05-31 19:39:10	1798
34	26	77	truck	天津市河西区尖山街道五一家园红星美凯龙国际家居...	2021-05-31 19:45:45	1042
35	20	68	truck	天津市河西区尖山街道惠程小区	2021-05-31 19:45:10	1780
36	27	78	truck	天津市河西区尖山街道电光大楼格调绮园	2021-05-31 19:46:27	497
37	39	87	truck	天津市河西区陈塘庄街道海河	2021-05-31 19:48:07	1705
38	10	54	truck	天津市西青区李七庄街道滨水西道延长线	2021-05-31 19:25:43	975
39	40	88	truck	天津市西青区李七庄街道滨水西道延长线	2021-05-31 19:27:11	22
40	41	89	truck	天津市西青区精武镇卓越北路天津工业大学新校区	2021-05-31 19:17:40	871

Figure 6. Driving history data storage map

图 6. 行车历史数据存储图

Table 1. Video storage table
表 1. 视频存储表

字段名	数据类型	长度	键属性	描述
video_id	String	10	主键	行车视频 ID
video_path	String	100		视频存储路径
gps	String	100		开始录制视频的位置
driver	String	5		驾驶员
video_startTime	Date	255		开始录制视频的时间

Table 2. Video processing table
表 2. 视频处理表

字段名	数据类型	长度	键属性	描述
process_id	String	10	主键	处理操作 ID
video_id	String	10	外键	行车视频 ID
detect_class	String	10		目标类别
detect_frame	Int	5		检测到该目标时的帧数
detect_startGPS	String	100		检测到该目标时的位置
detect_startTime	Date	255		检测到该目标时的时间

视频存储表用于存储行车视频相关信息，而视频处理表用于存储在 2.4 节整合好的四个数据集合信息，两张表通过 video_id 外键相关联。

3. 实验结果分析

为了更好的评估本文提出的模型，本文在真实的车辆上安装了行车记录仪来采集真实环境中的行车视频数据，如图 7 所示。行车记录仪品牌为第一现场，型号 D-269PRO，前置录像画面视角为 170°，视频分辨率为 1080P，运行内存为 1 GB。该设备每隔 3s 会记录一次行车位置并且具有夜间录像补光功能。本文录制的行车视频数据累计 28 小时，包含了白天以及晚上的行车视频。



Figure 7. Driving recorder installation location
图 7. 行车记录仪安装位置

本文采用了 Nvidia Jetson Nano [7]开发者套件作为开发平台,该设备是一款专为创客、学习者和开发者打造的小型 AI 计算机,如图 8 所示。Jetson Nano 使用了四核 64 位 ARM CPU 与 128 核集成 Nvidia GPU,达到了 472 GFLOPS 的计算性能。并且 Jetson Nano 附带的 Nvidia TensorRT [8]加速器库在多种场景下都极具实时性。

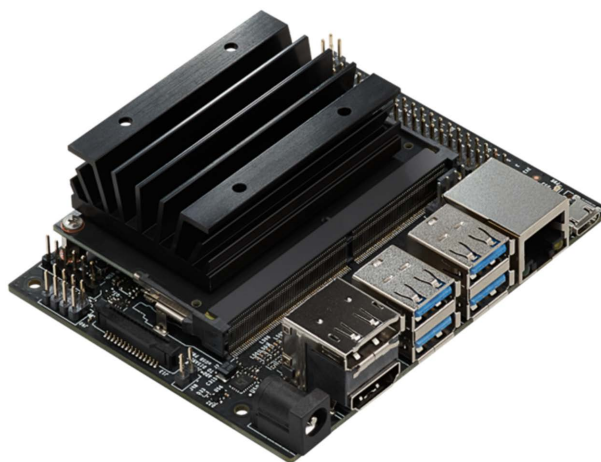


Figure 8. Jetson Nano developer kit
图 8. Jetson Nano 开发者套件

为了对该模型进行性能评估,本文进行了 5 次性能测试。每次测试保证只有该模型在运行,没有其他因素影响性能测试,例如:其他应用占用 CPU 资源或者 GPU 显存。实验结果为 5 次性能测试结果的平均值。详情见表 3。

Table 3. Performance test experiment results
表 3. 性能测试实验结果

性能指标	实验结果	标准差
帧率	5.45	0.02
错误率	7.3%	0.14
响应时间	120.3 ms	0.75
CPU 占用率	43.3%	0.02
内存使用率	68.3%	0.00

从表 3 的实验结果可以看出,系统运行平均帧率为 5.45,对于 Jetson Nano 小型计算机来说,已经达到了一个可观的性能。除此之外,响应时间达到 120.3 ms,证明可以达到用户的等待范围,提供一个良好的交互能力;CPU 占用率为 43.3%,证明可以轻松运行并处理当前运行的程序或者后台;内存使用率为 68.3%,证明还有充分的内存空间来处理其他任务;错误率只有 7.3%,证明本模型的准确性。综上所述,本文所提出的模型解决了历史数据提取任务,并且性能测试结果证明了其可行性。

4. 结论

本文为车内人机交互系统设计了一个行车历史数据提取模型,目的是解决车内 HMI 在提供个性化服务上的数据缺失问题。其次,为了验证所提出的模型,本文介绍了行车视频数据的采集以及性能实验所采用的硬件环境及软件环境。最后性能实验的结果证明了本文所提出的行车历史数据提取模型的可行性

以及准确性，为后续研究提供了理论基础和技术支持。

参考文献

- [1] 刘林峰. 基于智能化车载交互的语音识别系统的建立与设计[D]: [硕士学位论文]. 北京: 中国地质大学, 2014.
- [2] 莫宇凡. 基于情境意识的车载人机交互系统设计与研究[D]: [硕士学位论文]. 成都: 西南交通大学, 2020. <https://doi.org/10.27414/d.cnki.gxnju.2020.000719>
- [3] Smith, R. (2007) An Overview of the Tesseract OCR Engine. *Ninth International Conference on Document Analysis and Recognition*, Curitiba, 23-26 September 2007, 629-633. <https://doi.org/10.1109/ICDAR.2007.4376991>
- [4] Ren, S., He, K., Girshick, R., *et al.* (2017) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **39**, 1137-1149. <https://doi.org/10.1109/TPAMI.2016.2577031>
- [5] Greff, K., Srivastava, R.K., Koutník, J., *et al.* (2016) LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, **28**, 2222-2232. <https://doi.org/10.1109/TNNLS.2016.2582924>
- [6] DuBois, P. (2008) MySQL. Pearson Education, London.
- [7] Cass, S. (2020) Nvidia Makes It Easy to Embed AI: The Jetson Nano Packs a Lot of Machine-Learning Power into DIY Projects-[Hands on]. *IEEE Spectrum*, **57**, 14-16. <https://doi.org/10.1109/MSPEC.2020.9126102>
- [8] Vanholder, H. (2016) Efficient Inference with Tensorrt. *GPU Technology Conference*, Silicon Valley, 4-7 April 2016, 2.