软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2024,35(6):2753−2774 [doi: 10.13328/j.cnki.jos.006836]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

IATG: 基于解释分析的自动驾驶软件测试方法*

谢瑞麟[1], 崔展齐[1], 陈 翔[2], 郑丽伟[1]

[1](北京信息科技大学 计算机学院, 北京 100101)

[2](南通大学 信息科学技术学院, 江苏 南通 226019)

通信作者: 崔展齐, E-mail: czq@bistu.edu.cn

摘 要: 以深度神经网络 (deep neural network, DNN) 为基础构建的自动驾驶软件已成为最常见的自动驾驶软件解决方案. 与传统软件一样, DNN 也会产生不正确输出或意想不到的行为, 基于 DNN 的自动驾驶软件已经导致多起严重事故, 严重威胁生命和财产安全. 如何有效测试基于 DNN 的自动驾驶软件已成为亟需解决的问题. 由于 DNN 的行为难以预测和被人类理解, 传统的软件测试方法难以适用. 现有的自动驾驶软件测试方法通常对原始图片加入像素级的扰动或对图片整体进行修改来生成测试数据, 所生成的测试数据通常与现实世界差异较大, 所进行扰动的方式也难以被人类理解. 为解决上述问题, 提出测试数据生成方法 IATG (interpretability-analysis-based test data generation), 使用 DNN 的解释方法获取自动驾驶软件所做出决策的视觉解释, 选择原始图像中对决策产生重要影响的物体, 通过将其替换为语义相同的其他物体来生成测试数据, 使生成的测试数据更加接近真实图像, 其过程也更易于理解. 转向角预测模型是自动驾驶软件决策模块重要组成部分, 以此类模型为例进行实验, 结果表明解释方法的引入有效增强 IATG 对转向角预测模型的误导能力. 此外, 在误导角度相同时 IATG 所生成测试数据比 DeepTest 更加接近真实图像; 与 semSensFuzz 相比, IATG 具有更高误导能力, 且 IATG 中基于解释分析的重要物体选择技术可有效提高 semSensFuzz 的误导能力.

关键词: 深度神经网络; 自动驾驶软件; 解释方法; 软件测试

中图法分类号: TP311

中文引用格式: 谢瑞麟, 崔展齐, 陈翔, 郑丽伟. IATG: 基于解释分析的自动驾驶软件测试方法. 软件学报, 2024, 35(6): 2753–2774. http://www.jos.org.cn/1000-9825/6836.htm

英文引用格式: Xie RL, Cui ZQ, Chen X, Zheng LW. IATG: Interpretation-analysis-based Testing Method for Autonomous Driving Software. Ruan Jian Xue Bao/Journal of Software, 2024, 35(6): 2753–2774 (in Chinese). http://www.jos.org.cn/1000-9825/6836.htm

IATG: Interpretation-analysis-based Testing Method for Autonomous Driving Software

XIE Rui-Lin[1], CUI Zhan-Qi[1], CHEN Xiang[2], ZHENG Li-Wei[1]

[1](School of Computer Science, Beijing Information Science and Technology University, Beijing 100101, China)

[2](School of Information Science and Technology, Nantong University, Nantong 226019, China)

Abstract: Autonomous driving software based on deep neural network (DNN) has become the most popular solution. Like traditional software, DNN can also produce incorrect output or unexpected behaviors, and DNN-based autonomous driving software has caused serious accidents, which seriously threaten life and property safety. Therefore, how to effectively test DNN-based autonomous driving software has become an urgent problem. Since it is difficult to predict and understand the behavior of DNNs, traditional software testing methods are no longer applicable. Existing autonomous driving software testing methods are implemented byadding pixel-level

* 基金项目: 江苏省前沿引领技术基础研究专项 (BK202002001); 国家自然科学基金 (61702041); 北京信息科技大学“勤信人才”培育计划 (QXTCP C201906)
收稿时间: 2022-05-08; 修改时间: 2022-10-31; 采用时间: 2022-11-23; jos 在线出版时间: 2023-07-04
CNKI 网络首发时间: 2023-07-05

perturbations to original images or modifying the whole image to generate test data. The generated test data are quite different from the real images, and the perturbation-based methods are difficult to be understood. To solve the above problem, this study proposes a test data generation method, namely interpretability-analysis-based test data generation (IATG). Firstly, it uses the interpretation method for DNNs to generate visual explanations of decisions made by autonomous driving software and chooses objects in the original images that have significant impacts on the decisions. Then, it generates test data by replacing the chosen objects with other objects with the same semantics. The generated test data are more similar to the real image, and the process is more understandable. As an important part of the autonomous driving software's decision-making module, the steering angle prediction model is used to conduct experiments. Experimental results show that the introduction of the interpretation method effectively enhances the ability of IATG to mislead the steering angle prediction model. Furthermore, when the misleading angle is the same, the test data generated by IATG are more similar to the real image than DeepTest; IATG has a stronger misleading ability than semSensFuzz, and the interpretation analysis based important object selection method of IATG can effectively improve the misleading ability of semSensFuzz.

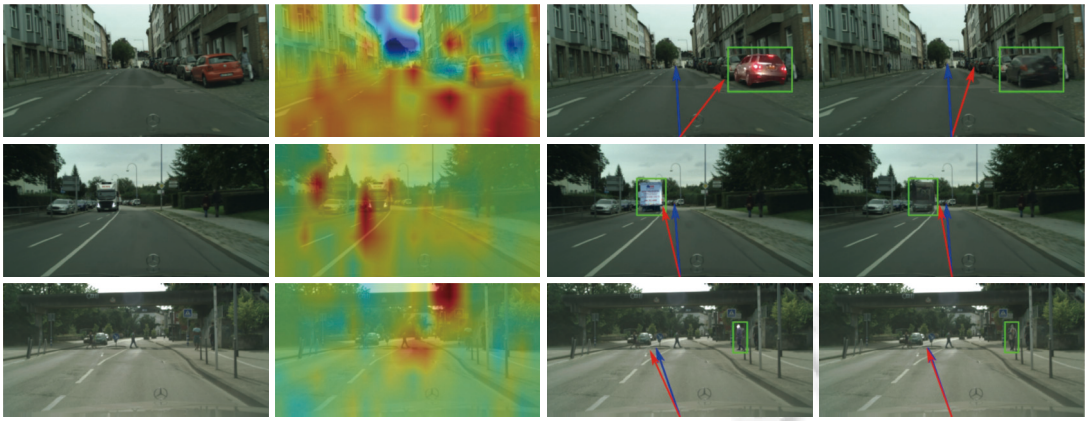
Key words: deep neural network (DNN); autonomous driving software; interpretation method; software testing

深度学习软件建立了类似人脑神经元的分层结构, 通过引入非线性的激活函数, 将输入的数据逐层转换为高维特征, 从而建立底层输入到高层语义的复杂映射关系[1], 可基于大规模数据集学习特定能力, 在某些特定任务上的准确性甚至已经超过了人类. 目前深度学习系统已经在计算机视觉、自然语言处理等领域取得了显著进展并得到了实际应用[2]. 作为深度学习软件的代表, 深度神经网络 (deep neural network, DNN) 在计算机视觉领域表现出卓越性能. 基于 DNN 的自动驾驶软件可对传感器采集的图像信息进行分析来做出驾驶决策, 已成为深度学习软件的典型应用之一[3]. 自动驾驶软件通常由感知模块、决策模块以及控制模块组成. 其中, 感知模块通过 2D 相机、雷达和 3D 传感器等多种传感设备监测车辆周围环境, 为自动驾驶软件的决策模块提供至关重要的环境信息, 决策模块收到由感知模块提供的多维度信息后, 依据当前环境信息进行驾驶行为决策, 并向车辆控制模块发送转向、加速和刹车等相关指令, 是自动驾驶软件中承担车辆行为决策任务的重要核心模块. 由于处理高维度特征信息的优异性能, DNN 被大量应用于构建决策模块.

与其他智能控制系统一样, 基于 DNN 的自动驾驶软件容易在特定环境或受到攻击时产生错误行为, 导致严重事故的发生. 如特斯拉自动驾驶汽车在 2021 年因自动驾驶软件的错误行为而导致严重事故[4]. 为减少此类事故的发生, 需要对自动驾驶汽车进行充分测试. 然而, 直接进行真车测试的成本过于高昂. 因此, 在使用真车测试之前通常会使用测试数据对相关模块进行模拟测试. 目前已经提出了多种基于图像的自动驾驶软件测试数据生成方法, 例如 DeepBillboard[5], DeepTest[6], DeepRoad[7]等. 但这些方法的生成过程缺乏解释、难以理解, 且所生成的测试数据和真实图像差异较大, 难以检测自动驾驶软件对相同场景做出不一致行为所导致的错误.

为解决上述问题, 本文提出了基于解释分析的自动驾驶软件测试数据生成方法 IATG (interpretability-analysis-based test data generation), 利用 DNN 解释方法生成 DNN 模型所做出决策的视觉解释, 选择原始测试数据图像中对决策模块产生重要影响的物体, 并通过基于真实图像和基于图像翻译两种替换策略将选择的重要物体替换为语义相同[8]的其他物体以生成测试数据, 使所生成的测试数据更加接近真实图像. 转向角控制是最基础的驾驶行为之一, 对决策模块中转向角预测模型的测试受到广泛关注. 本文对转向角预测模型进行测试的实验结果表明, IATG 所生成的测试数据能有效对决策模块产生误导, 且和 DeepTest 相比更加接近真实图像; 与 semSensFuzz[9]相比, IATG 则具有更高的误导能力, 且 IATG 中基于解释分析的重要物体选择方法可有效提高 semSensFuzz 生成测试数据的误导能力.

IATG 使用两种物体替换策略生成测试数据的示例如图 1 所示. 图 1 中第 1、2 列分别是原始图像和用解释方法生成的决策视觉解释 (以下称为热图). 第 3 列是基于真实图像的物体替换策略所生成的测试数据, 第 4 列是基于图像翻译的物体替换策略所生成的测试数据. 第 3、4 列中的蓝色箭头和红色箭头分别是 DNN 模型对原始图像和所生成测试数据的转向角预测结果, 其中绿色矩形框中的物体为经过解释方法进行选择并替换的物体. 从图 1 中的示例可以看出, IATG 的物体替换方法虽然只对原始图像中的小面积区域进行修改, 但依然可对 DNN 模型产生明显误导.



(a) 原始图像 (b) 视觉解释 (热图) (c) 基于真实图像的替换 (d) 基于图像翻译的替换

图 1 IATG 生成测试数据示例

本文的主要贡献总结如下.

● 提出了基于解释分析的自动驾驶软件测试数据生成方法 IATG, 其生成的测试数据能有效检测自动驾驶软件决策模块对相同场景做出不一致行为导致的错误.

● 基于所提出的方法实现了原型工具, 在 Cityscapes 数据集上就 IATG 生成测试数据的误导能力和真实性与 DeepTest 和 semSensFuzz 进行了对比, 并将基于解释分析的重要物体选择方法应用于 semSensFuzz, 以提高其生成测试数据的误导能力.

本文第 1 节介绍了相关研究背景. 第 2 节详细介绍所提出的基于解释分析的自动驾驶软件测试数据生成方法 IATG. 第 3 节介绍了实验设计, 并对实验结果进行了分析和讨论. 第 4 节进行有效性分析. 第 5 节介绍了相关工作并分析了 IATG 的创新性. 最后第 6 节总结全文并对未来工作进行展望.

1 研究背景

1.1 自动驾驶软件

自动驾驶软件的核心功能是捕获周围环境信息并加以分析, 以做出控制车辆驾驶行为的决策. 图 2 为一个基于深度学习的自动驾驶软件示例, 其由 2D 相机、雷达和 3D 传感器组成的传感器阵列捕获车辆所处的环境信息, 输入给用于决策的深度学习模型, 以供其对环境信息进行计算后形成转向角、加速控制和制动等驾驶行为决策[10]. 作为最基础的驾驶行为之一, 转向角预测受到了广泛关注[11,12]. 自英伟达发布使用卷积神经网络 (convolutional neural network, CNN) 进行转向角预测的 Dave[13]框架以来, CNN 成为最为常用的转向角预测模型. 其通过输出 $1/r$ 来表示转向角预测结果, 其中 r 为转弯半径, r 为负值表示左转, 正值表示右转, 当 r 的绝对值为一个极大值时则表示直行. 为方便理解, 本文将其转换为角度 (角度 = $180/r/\pi$) 来表示转向角预测结果.

CNN 是一种包含卷积操作层和前馈神经网络的 DNN 模型, 可以有效提取和分析图像特征, 被广泛应用于计算机视觉领域. 受益于参数共享和稀疏连接, CNN 非常适合处理图像数据. 其特有的卷积层结构由于共享权重的特性极大减少可训练权重的数量, 进而降低训练成本, CNN 的工作流程也更接近人类的视觉系统. 卷积层是 CNN 的关键组成部分, 它使用卷积核对前一层的输出进行卷积, 将特征提取为特征图并转递给后续层, 最后输入全连接层输出最终预测结果. 作为自动驾驶软件的关键组件, 对基于 CNN 的转向角预测模型进行充分测试是保障自动驾驶软件安全性和健壮性的重要手段.

1.2 针对转向角预测模型的测试数据生成技术

在针对自动驾驶转向角预测模型的测试数据生成的相关研究中, 常用方法是对在真实世界采集到的原始图片加入像素级扰动或对图片整体进行修改. 其中, 对图片进行像素级扰动的方法常使用基于梯度下降的白盒方法选择被修改像素. 然而, 现实世界中难以出现通过像素扰动所产生的图像. 即使可能由于图像采集设备缺陷而使输入

图像产生噪点等类似像素级扰动的情况, 所导致的错误在某种程度上也应该归咎于图像采集设备缺陷, 不能完全认为是自动驾驶软件中 DNN 模型缺陷导致. 而对图片整体进行修改的方法, 如 DeepTest[6]使用旋转、剪切、拉伸等方法来产生图像, 但产生的图像同样与真实图像差异较大. 添加雨天、雾天、亮度等天气效果的方法, 则只需要将装有图像和转向角采集设备的车辆由人类驾驶员在不同天气条件下的相同道路上驾驶便能采集到大量真实带标签的图像样本 (例如 Cityscapes[14]、BDD100K[15]等公开的大型自动驾驶数据集就是使用类似方法采集的). 本文将图像中实体的数量、语义及实体之间的位置关系称为场景 [16], 不同图像中若物体外观存在差异但数量、语义和位置相同, 则属于同一场景.

图 2 基于深度学习的自动驾驶软件

对于自动驾驶转向角预测模型的测试, 获取更多来自真实世界的测试数据并不困难. 其挑战在于难以检测模型由于对于相同场景产生不一致行为而导致的错误. 而此类错误是造成严重事故的常见原因之一. 例如, 2021 年在美国底特律, 一辆带有自动驾驶软件的特斯拉汽车因为误将一辆白色货箱的货车识别为天空而与之相撞 [4], 其搭载的自动驾驶软件之前在相同路段未发生过将货车误识别为其他物体的错误. 类似地, 2016 年和 2019 年在美国佛罗里达州也发生了两起因自动驾驶软件对于相同场景产生不一致行为而导致的严重事故 [17,18]. 在面对场景相同的不同图像时, 转向角预测模型应当产生一致或相似的结果. 使用场景相同的不同图像作为测试数据可有效检测出转向角预测模型的不一致行为. 如图 1 中第 1 行的第 1、3 张图像, 使用了外观不同的物体来描述同一场景, 但所使用的转向角预测模型却产生了差异较大的预测结果, 表明该模型内部存在缺陷.

1.3 DNN 的解释分析

以 DNN 为代表的深度学习技术在图像识别、语音识别和自然语言处理等应用领域取得了巨大进展, 被广泛地应用到一些重要的现实任务中, 例如自动驾驶 [19]、人脸识别 [20]、恶意软件检测 [21]和智慧医疗分析 [22]等. 在某些领域中, 深度学习模型的表现已经达到甚至超过了人类水平. 然而深度学习模型的学习和预测过程与传统的机器学习模型相比更不透明, 其根据样本学习特征以及做出预测的行为缺乏可解释性, 即使 DNN 模型的测试精度和其他性能指标已经达到了极高水平, 一个不可解释和被人理解的系统依然很难被人信任.

深度学习系统的行为难以预测且缺乏可解释性使其理论研究和应用发展受到严重阻碍. 近年来, 深度学习的可解释性问题引起了广泛关注 [23]. 其中, DNN 模型的可解释性是指使 DNN 模型具有自我解释或可外部解释其行为的能力或性质, 使人能理解和预测其行为. 常见的 DNN 解释方法有基于特征反演的 Guide Inversion[24]、基于类激活映射的 CAM[25]和基于梯度加权类激活映射的 Grad-CAM[26]等.

IATG 使用 Grad-CAM 作为 CNN 模型的解释方法. 给定一个输入样本, Grad-CAM 首先计算预测结果相对于最后一个卷积层中每一个特征图的梯度并对梯度进行全局平均池化, 以获得每个特征图的重要性权重. 然后, 基于重要性权重计算特征图的加权激活, 以获得一个粗粒度的梯度加权类激活图, 用于定位输入样本中具有类判别性的重要区域. 与 CAM 相比, Grad-CAM 无需修改网络架构或重训练模型, 避免了因修改模型或重训练带来的精度损失, 可适用于不同任务和结构的 CNN 模型.

2 基于解释分析的自动驾驶软件测试数据生成方法

IATG 的目标是基于解释分析从现有的原始图像生成测试数据, 以扩增测试集并增强测试充分性, 使其误导基于 CNN 的自动驾驶转向角预测模型, 以检测自动驾驶转向角预测模型中的潜在缺陷. IATG 的工作流程如图 3 所示. 首先, IATG 使用 CNN 的解释方法生成决策的视觉解释, 以此从原始图像中选择对决策产生重要影响的物体, 并用基于真实图像或图像翻译方法获取语义相同物体将其替换以生成测试数据. 然后, 将原始图像和测试数据输入模型, 比较输出的转向角差异来检测 CNN 模型对场景相同的不同图像输出不一致的错误行为.

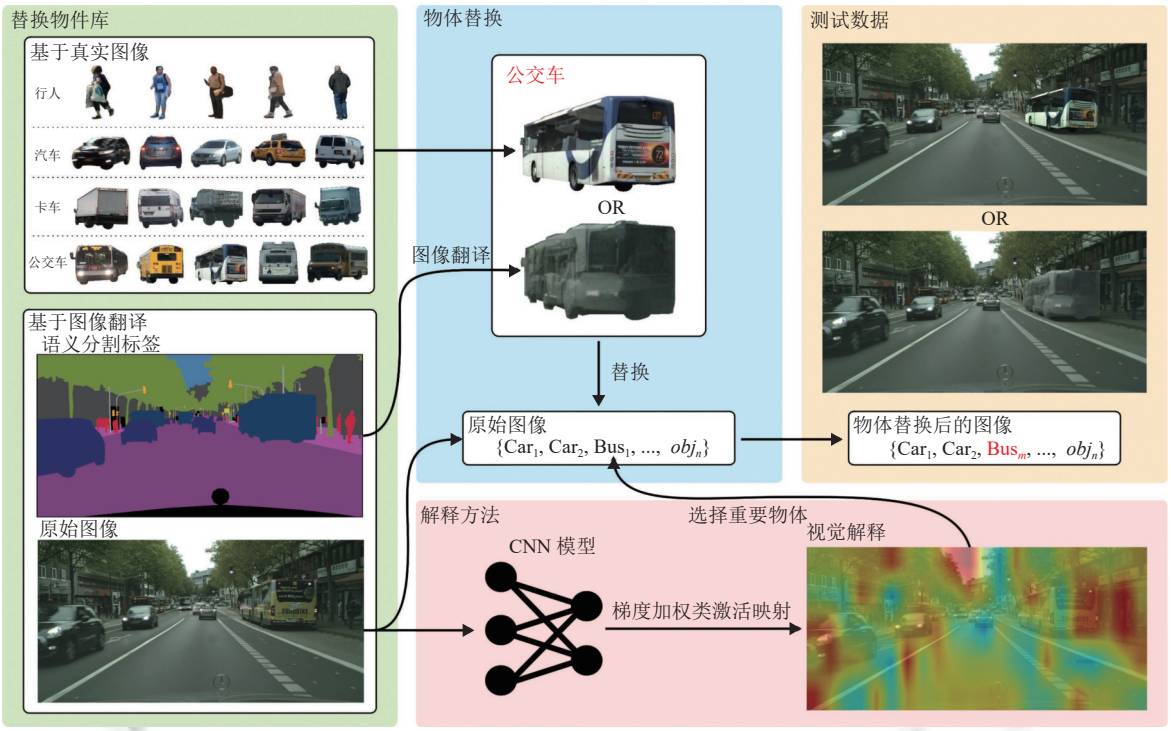


图 3 基于解释分析的自动驾驶软件测试数据生成方法流程图

2.1 转向角预测模型的解释分析

为检测转向角预测模型对相同场景产生不同预测结果的错误行为, IATG 将图像内的物体替换为语义相同的其他物体来生成测试数据, 以检测基于 CNN 的转向角预测模型缺陷. 然而, 一张原始图像可能包含数个甚至数十个物体, 盲目地替换大量物体或随机选择替换少量物体并不是明智的做法. 替换大量物体会增大测试数据与原始图像的差异, 降低测试数据的真实性; 随机选择替换少量物体则会降低测试数据误导转向角预测模型的能力. 因此, IATG 只替换少量对转向角预测模型输出结果贡献度较大的重要物体, 以确保测试数据在具有较强误导能力的同时, 尽量接近原始真实图像.

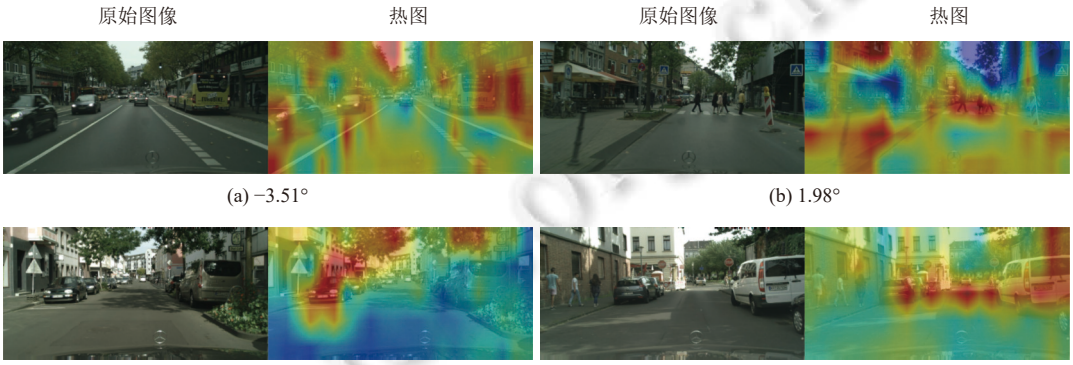
作为 CNN 的解释方法, Grad-CAM[26]适用于不同任务和结构的 CNN 模型, 在图像分类任务中只需要取最后一个卷积层的梯度信息即可计算像素对任意分类结果的贡献程度, 进而获得热图. 但在转向角预测这样的回归任务中, 模型输出为正值时代表角度偏右, 而输出为负值时代表角度偏左. 所以, 当某个特征图的梯度值为正时, 它将使模型倾向于右转; 当某个特征图的梯度值为负时, 它将使模型倾向于左转. 这意味着如果直接取梯度计算热图, 当模型预测结果为左转时, 热图中最高亮的区域将是对左转贡献度最低的区域, 不符合热图中越高亮的区域对预测结果越重要的性质, 因此当模型预测结果为左转时 IATG 无法使用热图选择重要物体.

为解决上述问题, IATG 在 Grad-CAM 方法的基础上, 在取梯度和计算特征图贡献值之间增加了一个步骤, 对

最后一层卷积层输出特征图的梯度进行如公式 (1) 所示的修改.

$$g' = \begin{cases} g, & \theta \geqslant 0 \\ -g, & \theta < 0 \end{cases} \tag{1}$$

其中, g 为特征图的原始梯度, g' 为经过修改后用于计算特征图贡献值的梯度, θ 为模型的转向角预测结果. 如公式 (1) 所示, 根据模型对于所输入图像的转向角预测结果的正负取向, 在转向角预测结果为负值, 也就是预测转向角为左转时, 对特征图的梯度取负. 经过这一步骤处理后, 无论模型最终的预测角度为左转还是右转, 生成的热图中越热的区域对模型的转向角决策越重要. 如图 4 所示为添加该步骤后使用 Grad-CAM 方法为转向角预测模型生成的部分热图. 其中左边两张为转向角预测为左转的原始图像及对应热图, 右边两张为转向角预测为右转的原始图像及对应热图.



(a) −3.51° (b) 1.98°

(c) −0.53° (d) 7.3°

图 4 使用 Grad-CAM 方法生成的热图

一个健壮性良好的模型在面对相同场景的不同图像时应当做出一致或相似的决策. 即使场景中的个别物体出现了变化, 但只要不改变物体的语义就不应该出现前后差异较大的决策. 例如, 模型不应该因为场景中黑色货车变成了一辆白色货车就改变原有的转向角预测结果. 如果模型出现了以上行为, 则可以认为检测到了模型中的缺陷, 或是发现了模型健壮性不佳的证据.

使用 Grad-CAM 方法生成的热图中, 越高亮的区域对模型的转向角预测结果贡献越高, 而被高亮区域覆盖的物体则是对转向角预测结果重要的物体. 对转向角预测结果贡献高的物体用语义相同的物体进行替换, 可在改变物体外观的同时不改变物体的语义, 来生成场景相同的不同图像. 由于被替换的是根据解释结果选择出的重要物体, 所以更有可能误导模型的预测结果.

然而, 如要在大量图像中替换部分物体, 同时使图像中其他物体保持不变, 在现实世界中是成本极高且几乎无法完成的任务. 使用已从现实世界采集到的原始图像, 从中选择重要物体并用语义相同物体替换后生成测试数据是较为可行的解决方案. 下面将介绍基于真实图像和图像翻译两种获取物体并进行替换的策略.

2.2 重要物体替换

利用解释方法分析并选择待替换物体后, 将使用替换物体库中的新物体替换原物体. 新物体应与原物体保持相同的语义并尽可能保证替换后图像的真实性. 替换物体库是一组物体及其语义标签的集合, IATG 使用两种策略获取物体并构建替换物体库: 基于真实图像构建和基于图像翻译构建. 基于真实图像构建策略使用从真实世界采集的图像, 按照语义分割标签信息提取真实物体. 而基于图像翻译构建策略则是使用图像翻译工具将语义分割标签翻译为图像, 然后从中提取所生成的物体. 构建替换物体库后, 可使用其中与被选择重要物体语义相同的物体进行替换.

重要物体替换的具体步骤如算法 1 所示, 算法的输入为原始图像 *orgImg*、转向角预测模型 M、替换物体数量 N 和替换物体库 *objLib*, 算法的输出为生成的测试数据图像 *newImg*. 首先使用解释方法 Grad-CAM 分析原始图

像 *orgImg* 并生成热图 *HeatMap* (第 1 行); 然后对原始图像 *orgImg* 进行物体分析以识别图像中的所有物体和语义和像素位置信息, 并将所有信息储存在所有物体信息 *allObjInfo* 中 (第 2 行); 之后根据所有物体信息 *allObjInfo* 和热图 *HeatMap* 对所有物体进行重要性分析并排序生成重要性排序 *importanceSort* (第 3 行); 根据重要性排序 *importanceSort* 从所有物体信息 *allObjInfo* 中选择最重要的前 *N* 个物体的信息并储存在重要物体信息 *objInfo* 中 (第 4 行); 用原始图像 *orgImg* 初始化测试数据图像 *testImg* 后, 对重要物体信息 *objInfo* 中的每一个物体, 先根据物体像素位置信息 *obj.location* 将其从测试数据图像 *testImg* 中删除, 再根据物体语义 *obj.semantic* 从替换物体库 *objLib* 中选择语义相同的物体替换到原来的位置 (第 5–9 行). 在对重要物体信息 *objInfo* 中的每一个物体都完成替换后, 输出生成的测试数据图像 *testImg* (第 10 行).

算法 1. 重要物体分析及替换.

输入: 原始图像 *orgImg*; 转向角预测模型 *M*; 替换物体数量 *N;* 替换物体库 *objLib*;
输出: 新测试数据图像 *newImg*.

1. *HeatMap* = GradCAM(*orgImg, M*)
2. *allObjInfo* = objAnalysis(*orgImg*)
3. *importanceSort* = importanceAnalysis(*allObjInfo, HeatMap*)
4. *objInfo* = objChoose(*allObjInfo, importanceSort, N*)
5. *testImg* = *orgImg*
6. **for** esch *obj* in *objInfo*:
7. *testImg* = objDel (*newImg, obj.location*)
8. *testImg* = objReplace(*newImg, obj.semantic, objLib*)
9. **end for**
10. **return** *testImg*

算法 1 中, 对物体进行重要性分析与排序需要计算物体的重要性值并以此为依据对其重要性进行排序. IATG 使用两种不同的重要性值计算方法, 具体计算方法将在第 2.2.1 节介绍. 替换物体库 *objLib* 的构建和替换方法在基于真实图像和基于图像翻译两种策略中并不相同, 具体方法将在第 2.2.2 节和第 2.2.3 节介绍.

2.2.1 物体重要性分析

为选出对转向角预测模型决策产生重要贡献的物体, 需要根据算法 1 第 1 行得到的热图分别对原始图像中每个物体计算其重要性值, 并以此作为选择重要物体的量化依据对所有物体进行重要性排序. 对物体重要性值的计算使用了两种方法, 分别为基于物体总贡献和基于物体平均贡献的重要性值, 具体计算方法如公式 (2) 和公式 (3) 所示.

令 $I=\{P_1,P_2,P_3,\ldots\}$ 为原始图像中像素的集合, $P_i\in I$ $(1\leqslant i\leqslant |I|)$ 为图像中的一个像素, $obj\subseteq I$ 为原始图像中的一个物体, $H(P_i)$ 为像素 P_i 在热图中对应的贡献值. 公式 (2) 定义了基于物体总贡献重要性值 K_{sum} 的计算方法, 其计算的是物体所有像素对转向角预测模型决策贡献值的总和.

$$K_{\text{sum}}(obj)=\sum\nolimits_{P_i\in obj}H(P_i) \tag{2}$$

公式 (3) 定义了基于物体平均贡献重要性值 K_{avg} 的计算方法, 其计算的是物体所有像素对转向角预测模型决策贡献值的算数平均值.

$$K_{\text{avg}}(obj)=\frac{\sum_{P_i\in obj}H(P_i)}{|I|} \tag{3}$$

2.2.2 基于真实图像的物体替换

此策略使用来自真实世界采集图像中提取的物体. 为保证所提取物体的完整性和语义正确性, 需要先生成真实图像的语义分割标签, 以将不同语义以及相同语义的不同物体与背景之间区分开来, 然后根据语义分割标签从

图像中提取真实物体并与其语义标签一同组成替换物体库, 以供物体替换使用.

使用此策略提取物体可保证其真实性和语义一致性, 但无法保证物体形状轮廓与原始物体的一致性, 直接使用所提取的物体进行替换可能出现无法完全覆盖原始物体区域的情况. 因此, 在使用真实图像提取的物体进行物体替换时, 需要先将原始物体从原始图像中删除, 并使用内容识别填充技术将物体删除后的区域填充为周边背景, 然后再使用真实图像提取的物体进行物体替换. 如图 5 所示为使用真实图像提取的物体进行物体替换的流程示例.

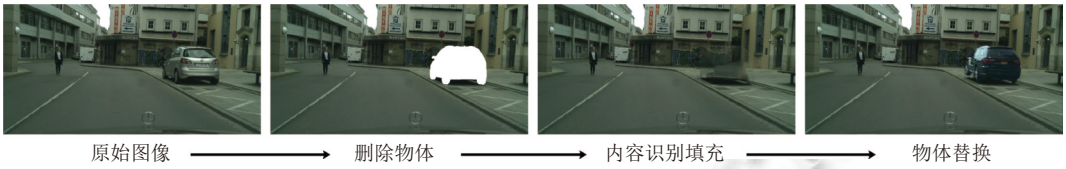


图 5 使用真实图像提取的物体进行物体替换

2.2.3 基于图像翻译的物体替换

图像翻译是指从一副图像到另一副图像的转换[27], 类似机器翻译将一种语言转换为另一种语言的过程. pix2pix[28]、CRN[29]和 pix2pixHD[30]等典型的图像翻译方法具有将语义分割标签转换为接近真实街景的图像、灰色图像转换为彩色图像、白天照片转换为黑夜等能力. 其中, pix2pixHD 甚至可将详细标注的语义分割标签直接翻译为接近真实图像的场景. 基于图像翻译的物体替换过程中, 把原始图像的语义分割标签输入图像翻译工具, 将语义分割标签翻译为与其对应图像, 所生成的图像中即包含了与待替换物体的轮廓、位置和语义相同的物体, 将其提取并与其语义标签一同组成替换物体库. 使用此策略提取的物体虽然不是真实物体, 但其形状轮廓与原始物体完全一致, 可以直接进行替换. 使用此策略所生成物体的质量取决于图像翻译工具的有效性. 如图 6 所示为使用图像翻译工具生成的物体进行替换的流程示例.

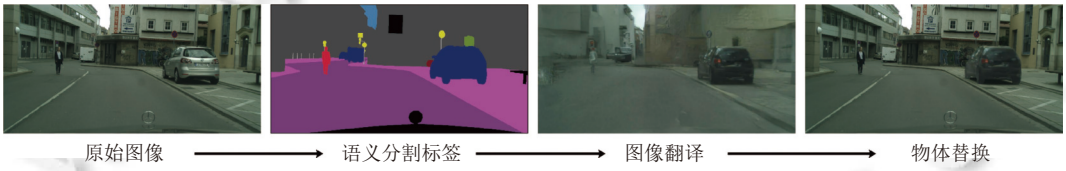


图 6 使用图像翻译工具生成的物体进行物体替换

3 实验设计与结果分析

3.1 实验设计

实验将重点关注自动驾驶软件决策模块中的转向角预测模型. 英伟达 DAVE[13]自动驾驶架构被广泛用于构建转向角预测模型[5−7], 实验对象采用使用该架构的预训练模型 (Steering angle visualizations, https://github.com/jacobgil/keras-steering-angle-visualizations). 实验使用 Cityscapes 数据集[14]的训练集作为原始图像集, 其中包含了 3 475 张真实图像. IATG 中基于图像翻译的物体替换策略使用 pix2pixHD 作为图像翻译工具, pix2pixHD 发布的预训练模型和开源代码使用 Cityscapes 数据集训练预训练模型. 为确保图像翻译工具的有效性, 实验将 Cityscapes 作为实验数据集. 实验过程中将 IATG 生成的测试数据输入目标模型获取预测结果, 并对所生成测试数据的质量进行评价. 所有测试都在 32 GB RAM、AMD 3700X 8-Core Processor 和英伟达 RTX 3070Ti GPU 的计算机上完成.

本文通过以下研究问题来验证所提出自动驾驶测试数据生成方法的有效性.

RQ1: 不同物体选择策略、替换策略、替换物体数量如何影响 IATG 生成测试数据的误导能力和学习感知图像块相似度 (learned perceptual image patch similarity, LPIPS)[31]?

RQ2: 是否使用解释方法, 对 IATG 生成测试数据误导能力的影响如何?

RQ3: 与其他测试数据生成方法相比, IATG 生成测试数据的误导能力和 LPIPS 值如何?

不同的参数设置将影响 IATG 生成测试数据的质量, 为此设计了 RQ1 用以讨论实验参数的设置. 其中, LPIPS 为用于评价测试数据与原始测试数据的相似程度的指标. 为验证 IATG 的有效性, 设计了 RQ2 和 RQ3 以验证和评价 IATG 的有效性和所生成的测试数据的质量.

3.2 构建替换物体库

实验使用 Cityscapes[14]和 BDD100K[15]两个数据集分别使用基于真实图像和基于图像翻译两种策略构建替换物体库.

3.2.1 基于真实图像构建替换物体库

Cityscapes[14]、BDD100K[15]等公开的大型自动驾驶数据集都提供了精细的语义分割标签, 详细标注了图像中不同物体以及背景的精细轮廓和语义类型. IATG 可使用语义分割标签将不同语义的物体从图像中提取出来. 被提取的物体应该具有基本的辨识度和完整性以保证语义一致性, 因此只保留像素大于 50×50, 并且不被其他物体遮挡或截断的物体. 对于从真实图像中提取物体的策略, 使用了 BDD100K[15]数据集作为提供真实图像的数据集, 其中包含 10 万段高清视频, 每个视频约有 1 200 帧. 对每个视频第 10 秒的关键帧进行采样, 得到 10 万张图片 (图片尺寸: 1280×720), 并进行详细的语义分割标注, 利用标签将不同语义的物体从原始图像中分离即可获取来自真实图像的物体.

如图 7 所示为从 BDD100K 数据集提取的物体构建的替换物体库示例, 其中左侧为 BDD100K 数据集的原始图片和语义分割标签示例, 右侧为使用原始图像按照不同语义标签提取完整物体及其语义标签构建的替换物体库示例. 由于基于真实图像提取的物体不能保证其轮廓形状与待替换物体相同, 在替换之前需要将待替换物体从原始图像中删除, 然后用内容识别填充方法将物体删除后的区域填充为周边背景, 并使用真实图像提取的物体进行物体替换. 实验使用 Adobe Photoshop 软件的内容识别填充和批处理功能填充物体删除后的背景区域.

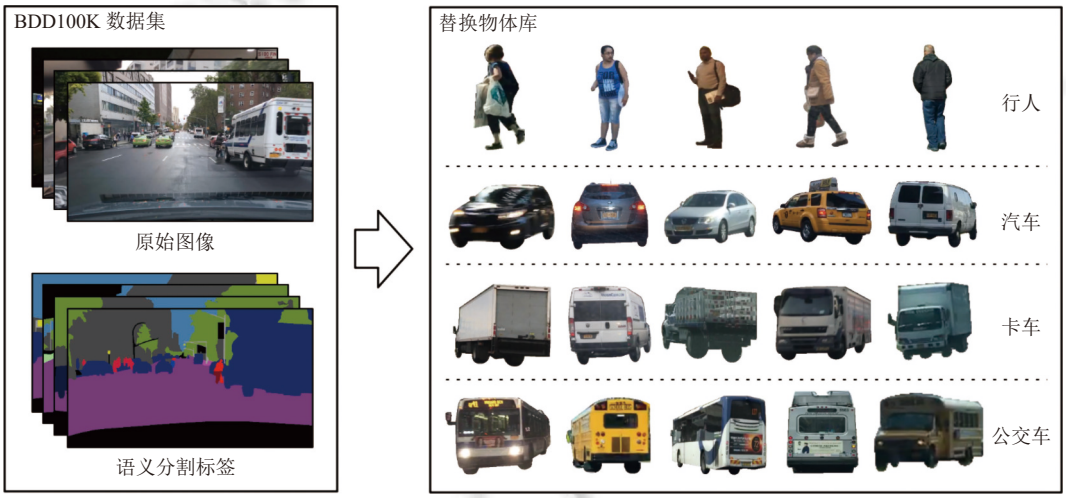


图 7 从 BDD100K 数据集提取的物体示例

3.2.2 基于图像翻译构建替换物体库

对于基于图像翻译的物体替换策略, 使用 pix2pixHD[30]发布的预训练模型和开源代码 (pix2pixHD, https://github.com/NVIDIA/pix2pixHD) 作为图像翻译工具, 并使用与预训练模型对应的 Cityscapes 数据集的训练集作为语义分割标签来源. 该训练集中有 3 475 张在城市道路采集的图像, 共有 19 个语义分割标签分类, 其中 8 个类具有详细的实例级标注. 将语义分割标签输入 pix2pixHD 的预训练模型, 可将语义分割标签翻译为与真实图像类似的图像. 由于 Cityscapes 数据集同时作为实验部分的原始图像来源, 所以可得到与待替换物体轮廓完全匹配的新物

体, 所生成图像中的物体及其对应的语义标签可直接作为替换物体库. 如图 8 所示为使用 pix2pixHD 生成的图像示例, 其中第 1、3 列为 Cityscapes 数据集中 4 张不同图像的语义分割标签, 第 2、4 列为 pix2pixHD 分别使用 4 张语义分割标签生成的场景图像.

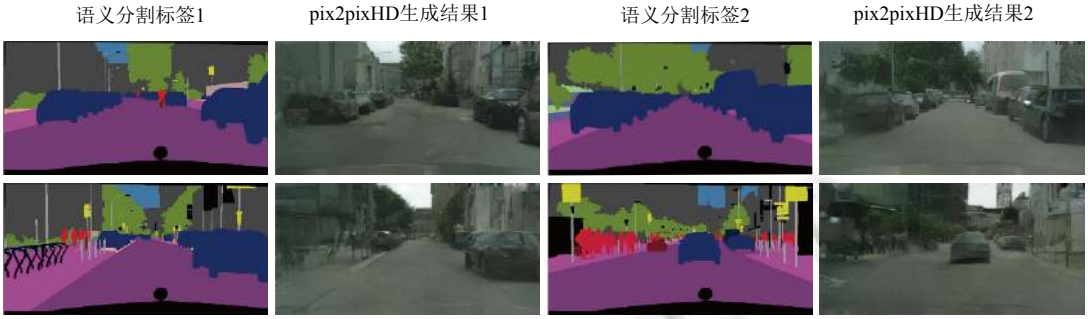


图 8 使用 pix2pixHD 生成的图像示例

3.3 测试数据质量度量

实验主要从两个方面评价测试数据的质量: 转向角预测的误导能力和与原始图像的接近程度.

3.3.1 转向角预测误导能力

考虑到驾驶员在相同场景做出的转向角判断也不会完全一致, 所以基于 CNN 的转向角预测模型在一定范围内的误差不能被认为其预测结果出现错误. 实验中通过设置恰当阈值的方式判断测试数据是否触发了转向角预测模型的错误行为: 当生成测试数据输入模型后的预测转向角与原始图像输入后的预测转向角之差 (以下称为误导角度) 大于阈值, 则认为成功误导并触发了模型的错误行为. 为了检测以上错误行为, 实验沿用了文献 [5] 中测试集对模型误导能力的指标 M_1 来度量其对模型的误导能力, 其具体计算方法如公式 (4) 所示.

令 $X=\{x_1,x_2,x_3,\ldots\}$ 为原始测试集, $x_i\in X$ $(1\leqslant i\leqslant |X|)$ 为其中一条测试数据, x_i' 为基于原始测试数据 x_i 生成的测试数据, $f(\cdot)$ 为转向角预测模型根据所输入数据输出的预测结果, α 为误导角度阈值.

$$M_1=\frac{\left|\{x_i|f(x_i')-f(x_i)>\alpha,0<i<|X|\}\right|}{|X|} \tag{4}$$

3.3.2 与真实图像的相似度

与真实图像的接近程度是指原始测试用数据添加扰动后所生成测试数据与原始测试数据的相似程度. 当原始测试数据为从现实世界采集的真实图像时, 所生成的测试数据与原始测试数据越相似, 可认为测试数据越接近真实图像.

学习感知图像块相似度 (LPIPS)[31]可用以评价测试数据与原始测试数据的相似程度. LPIPS 通过小型神经网络学习目标 DNN 模型中待评价图像和真实图像的距离到人类对其相似度评价的反向映射, 以评价待评价图像和真实图像间的感知相似度. LPIPS 比 SSIM[32], MSSIM[33]和 FSIM[34]等方法更符合人类对图像的感知情况. LPIPS 值越低表示两张图像越相似, 反之, 则差异越大. 为计算 LPIPS 值, 首先将原始图像和待评价图像分别输入目标 DNN 模型并计算图像之间的距离 d; 然后用 d 作为特征训练一个用于预测图像间相似度的小型神经网络, 训练所得神经网络的预测结果即为 LPIPS 值. 实验中使用预训练的 VGG 模型作为目标 DNN 模型, d 的具体计算方法如公式 (5)[31]所示.

$$d(x,x_0)=\sum_l\frac{1}{H_lW_l}\sum_{h,w}\left\|w_l\odot\left(\hat{y}^l_{hw}-\hat{y}^l_{0hw}\right)\right\|_2^2 \tag{5}$$

令 l 为目标 DNN 模型中的一层, 首先将真实图像 x 和待评价图片 x_0 分别输入目标模型, H_l 和 W_l 分别为 l 层特征堆栈的高度和宽度, 从 l 层提取特征堆栈 (feature stack) 并在通道维度中进行单位归一化 (unit-normalize) 得到 $\hat{y}^l_{hm}$ 和 $\hat{y}^l_{0hm}$; 然后使用权重向量 w_l 来放缩激活通道并计算 L_2 范数, 取平方后在特征堆栈的空间上求平均值, 并逐

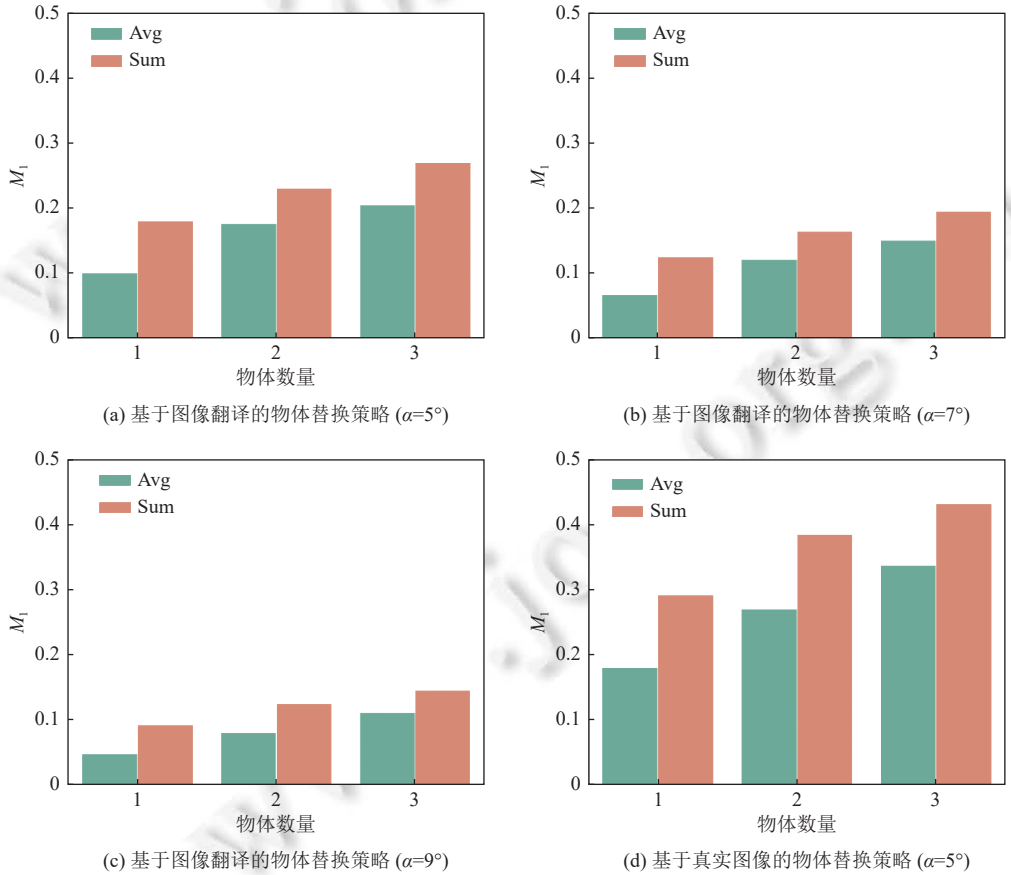
层求和.

3.4 RQ1 实验结果分析

为了研究不同参数设置对 IATG 误导能力的影响, IATG 首先使用解释方法获取热图并使用平均贡献度和总贡献度两种策略来选择被替换的重要物体, 然后使用基于真实图像和基于图像翻译的两种策略来替换所选择的重要物体. 实验评价了对于不同误导角度阈值 α, IATG 所生成测试数据的误导能力 M_1 及 LPIPS 值. 其中误导角度阈值 α 分别设置为 5°、7°和 9°. 假设某车身宽 1.8 m 的自动驾驶汽车以 50 km/h (常见城市道路限速) 的速度在 3.5 m 宽的车道中央行驶. 当转向角预测模型产生 5°、7°和 9°的转向角预测误差时, 将分别在 0.7 s、0.5 s 和 0.4 s 内使车辆偏离车道中心线并越过车道边界. 而人类驾驶员发现异常并采取转向角修正或车辆制动等措施进行干预的反应时间为 0.7 s 至 2.3 s[35]. 因此, 当出现以上转向角偏差时, 驾驶员将很难及时做出反应并采取干预措施.

两种物体替换策略在第 4.2 节进行了详细介绍. 物体平均贡献度和物体总贡献度是指使用解释方法获取热图后, 分别计算的物体重要性值 K_{avg} 和 K_{sum}. 通过两种不同的重要性值计算方式计算图像中所有物体的贡献度, 并进行排序, 选取排名靠前且物体的长和宽不小于 50 像素的物体作为待替换物体. 所选择的物体是对模型转向角预测造成较大影响的物体, 因此不选择面积过小的物体.

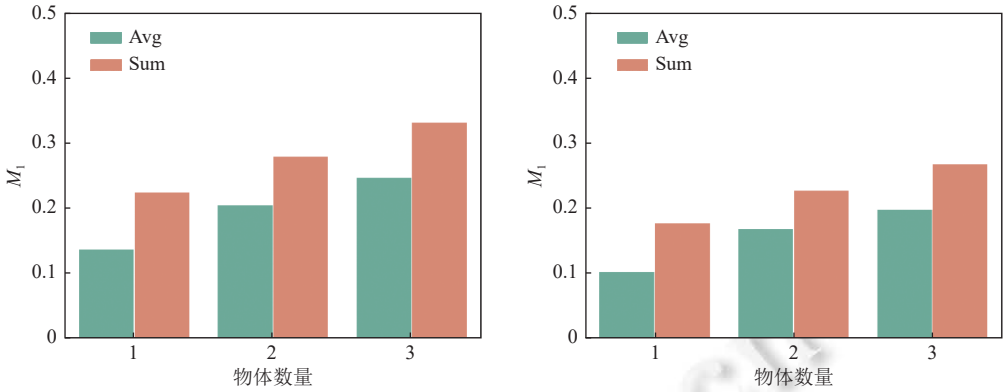
不同参数设置和策略下的误导能力对比实验结果如图 9 所示. 图 9 中, 纵轴为触发错误的测试数据占所有测试数据的比例, 横轴为替换物体数量. 图 9(a)、图 9(b) 和图 9(c) 为使用基于图像翻译的物体替换策略, 图 9(d)、图 9(e) 和图 9(f) 为使用基于真实图像的物体替换策略, 其中 α 代表评价指标 M_1 的误导角度阈值 (误导角度大于阈值则认为触发错误). 图 9 中 Avg 表示使用物体平均贡献度 (K_{avg}) 选择物体, Sum 代表使用物体总贡献度 (K_{sum}) 选择物体.



(a) 基于图像翻译的物体替换策略 (α=5°)　(b) 基于图像翻译的物体替换策略 (α=7°)

(c) 基于图像翻译的物体替换策略 (α=9°)　(d) 基于真实图像的物体替换策略 (α=5°)

图 9　不同策略和参数生成测试数据误导能力对比



(e) 基于真实图像的物体替换策略 (α=7°)　　(f) 基于真实图像的物体替换策略 (α=9°)

图 9　不同策略和参数生成测试数据误导能力对比 (续)

不同的参数设置和策略生成的测试数据数量并不相同, 这是因为有的参数设置和策略将无法从部分原始图像中选出足够的待替换物体. 例如设置替换物体数量为 3, 而原始图像内的可供选择的物体数量小于 3. 其中, 使用真实物体替换策略、设置替换物体数量为 3, 使用总贡献度方式计算物体贡献度的参数和策略设置所生成测试数据最少, 为 2 475 张图像.

由图 9 中可以看出, 在只改变单一因素而其他策略和参数设置相同的情况下, 实验结果都遵循了以下的现象.

(1) 使用基于真实图像的物体替换策略, 比使用基于图像翻译的物体替换策略的误导成功率更高.

(2) 使用物体总贡献度选择待替换物体, 比使用物体平均贡献度误导成功率更高.

(3) 替换物体数量越多, 误导成功率越高.

从真实图像中提取的物体通常和待替换物体的轮廓形状存在一定差异, 而图像翻译工具生成的物体与待替换物体的轮廓完全相同, 所以真实物体与待替换物体的差异性更大, 能更有效地误导模型, 从而导致产生现象 (1). 使用物体总贡献计算贡献度的方法选择待替换物体的策略所选择的物体, 其在面积上通常比使用物体平均贡献计算贡献度的策略所选择的物体更大, 所以物体替换后与原始图像的差异更大, 能更有效地误导模型, 从而导致产生现象 (2). 相似地, 替换物体数量越多, 则物体替换后与原始图像的差异越大, 能更有效地误导模型, 从而导致产生现象 (3).

不同参数和策略设置的 LPIPS 值对比实验结果如后文图 10 所示. 其中, 纵轴为 LPIPS 值, 横轴为替换物体数量. 图 10(a) 为使用基于图像翻译的物体替换策略, 图 10(b) 为使用基于真实图像的物体替换策略. 如图 10 所示, IATG 在不同策略和参数设置下生成的测试数据除了少量个例以外, LPIPS 值均小于 0.05 (LPIPS 的值域为 [0, 1), 越接近 0 则表示测试数据越接近原始真实图像). 说明 IATG 在以上策略和参数设置下均能生成真实性较高的测试数据. 其中, 替换物体数量越少, 真实性越高; 根据物体平均贡献度选择待替换物体的策略, 比使用物体总贡献度的真实性高; 使用基于图像翻译的物体替换策略, 比使用基于真实图像的物体替换策略更加接近原始真实图像.

针对 RQ1 的结论: 将替换物体数量设置为 3 个, 使用物体总贡献度计算贡献度的方法选择待替换物体, 并使用真实物体进行替换的策略将得到误导能力最强的测试数据集. 将替换物体数量设置为 1 个, 使用 物体平均贡献度计算贡献度的方法选择待替换物体, 并使用基于图像翻译的物体替换策略将得到更接近 真实图像的测试数据集.

3.5　RQ2 实验结果分析

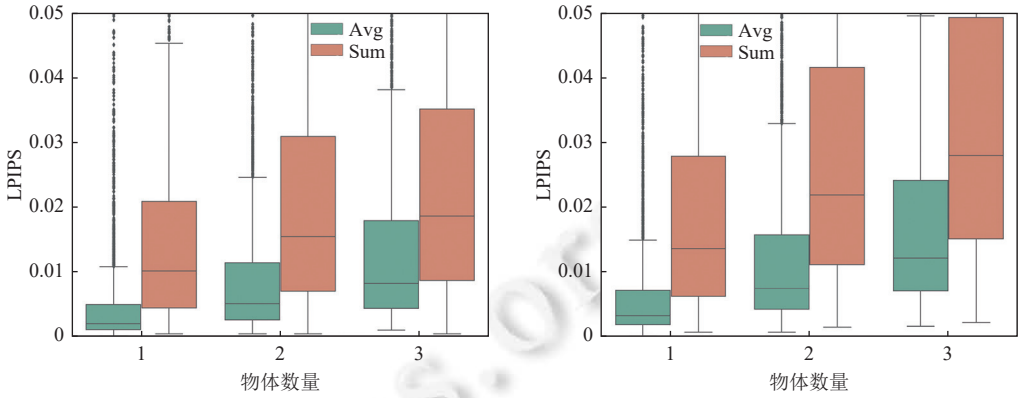
为了研究使用解释方法选择被替换物体对 IATG 的影响, 使用 IATG 方法和随机选择被替换物体的方法进行对比实验. 其中, 随机选择被替换物体的方法为 IATG 的基础上, 使用随机选择的方法替换基于解释分析选择被替换物体的方法 (以下称为 $\text{IATG}_{\text{rand}}$). 基于第 3.4 节的实验结论, 使用了以下两种参数设置和策略作为被比较

的对象.

S1 (最大化误导能力): 根据物体总贡献度选择 3 个被替换物体, 及基于真实图像的物体替换策略.

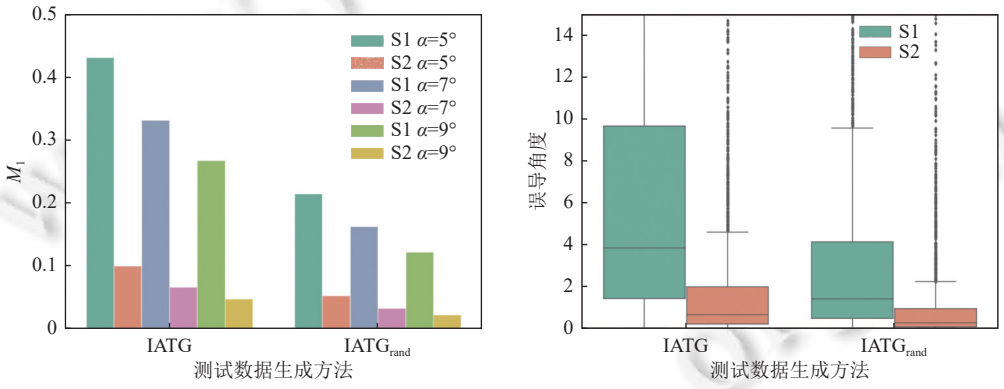
S2 (最大化真实性): 根据物体平均贡献度选择 1 个被替换物体, 及基于图像翻译的物体替换策略.

分别按照 S1 和 S2 两种参数设置, 使用 IATG 和 $\text{IATG}_{\text{rand}}$ 进行实验. 实验结果如图 11 所示, 其中图 11(a) 为 α 值不同时, 两种方法在两种参数设置下所生成测试数据集成功误导目标 CNN 模型比例的柱状图; 图 11(b) 为两种方法在两种参数设置时所生成测试数据集中所有图片的误导角度箱型图.



(a) 基于图像翻译的物体替换策略 (b) 基于真实图像的物体替换策略

图 10 不同策略和参数的 LPIPS 值对比



(a) M_1 (b) 总体误导角度

图 11 不同测试数据生成方法的误导能力对比

由图 11(a) 可见, 在设置的策略和 α 值和不同时, IATG 方法均比 $\text{IATG}_{\text{rand}}$ 更能有效误导目标 CNN 模型. 其中, 设置参数策略为 S1 和 S2 时, IATG 生成测试数据的成功误导比例比 $\text{IATG}_{\text{rand}}$ 分别高 2.1 倍和 1.9 倍. 实验结果表明, 解释方法的引入提高了 IATG 对 CNN 模型的误导能力, 参数策略 S1 中替换物体数量为 3 个物体, 相比 S2 替换 1 个物体更能发挥解释方法能选择影响 CNN 模型决策重要物体的优势.

由图 11(b) 可见, 当参数设置不同时, IATG 所生成的测试数据集在整体上比 $\text{IATG}_{\text{rand}}$ 能导致更大的误导角度. IATG 生成测试数据误导角度的中位数、上四分位数和上边缘均比 $\text{IATG}_{\text{rand}}$ 高 1 倍以上. 这表明引入了解释方法的 IATG 能地误导待测 CNN 模型产生更大的误导角度.

针对 RQ2 的结论: 解释方法的引入提高了 IATG 对转向角预测模型的误导能力, 所生成测试数据的成功误导比例和误导角度均高于随机选择被替换物体的方法.

3.6 RQ3 实验结果分析

作为基于图像的自动驾驶软件测试数据生成方法, DeepTest[6]常被用作转向角预测模型测试研究的对比基线[5−7,36], 而 semSensFuzz[9]同样关注所生成测试数据的真实性和对深度学习模型的误导能力. 为评价 IATG 生成测试数据的有效性, 我们从 DeepTest 和 semSensFuzz 的 GitHub 仓库 (DeepTest, https://github.com/ARiSE-Lab/deepTest), DeepTest 中获取其实现代码用于生成测试数据, 并与 IATG 生成的测试数据进行了比较. 实验中使用 DeepTest 定义的 6 种测试数据生成方法 (brightness, contrast, rotation, scale, shear 和 translation) 生成了 17 850 张测试数据, 使用 semSensFuzz 定义的两种图像数据变异方法(改变物体颜色和添加物体) 生成了 150 000 张测试数据, 其中改变物体颜色的方法只改变原始图像中某一物体的颜色, 并不会改变物体的语义和位置, 可认为其生成的图像与原始图像属于同一场景而具有相同的测试预言. 而添加物体则会向原始图像中添加一个新物体, 所添加的物体和原有物体之间将产生新的空间关系, 无法认为其生成的图像与原始图像属于同一场景, 新测试数据将具有不同的测试预言. 因此, IATG 将和 semSensFuzz 所生成的 15 万张新测试数据比较 LPIPS 值, 但只与其中 5 万张使用改变物体颜色方法生成的新测试数据比较误导角度. DeepTest 和 semSensFuzz 的参数均设置为默认值, IATG 的参数设置沿用第 3.5 节的设置. 实验结果如图 12 所示, 其中图 12 为 IATG、DeepTest 和 semSensFuzz 使用改变物体颜色方法所生成测试数据的误导角度对比箱型图; 图 13 为 IATG、DeepTest 和 semSensFuzz 使用全部两种方法所生成测试数据的 LPIPS 值对比的箱型图.

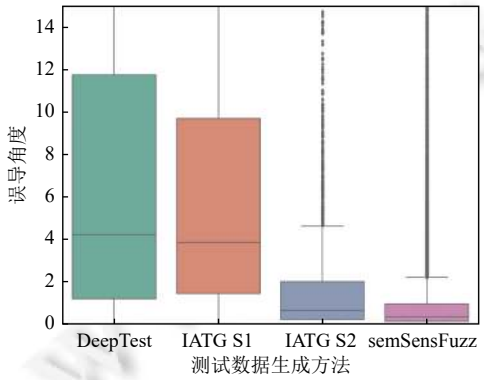


图 12 不同测试数据生成方法的误导角度对比

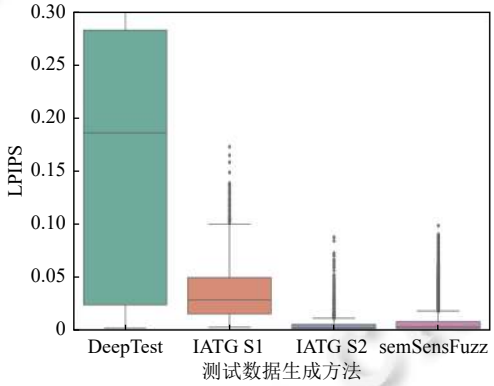


图 13 不同测试数据生成方法的 LPIPS 值对比

与 DeepTest 相比, 由图 12 可见, 当参数设置为 S1 时, IATG 误导角度的上四分位数比 DeepTest 小 2.1°, 而中位数则仅比 DeepTest 小 0.4°. 这表明 IATG 的误导能力略低于 DeepTest. 但由图 13 可见, 当参数设置为 S1 和 S2 时, IATG 的 LPIPS 值中位数分别为 DeepTest 的 15.03% 和 1.04% (LPIPS 值越接近 0 则表示生成的图像越接近真实的原始图像). 这表明 IATG 所生成的测试数据相比 DeepTest 更加接近原始图像, 即更加接近真实图像. 这是由于 IATG 只替换图像中的部分物体并不会改变图像的其他区域, 而 DeepTest 的方法通常会修改整张图像, 这使得 IATG 生成的测试数据将更接近原始图像.

与 semSensFuzz 相比, 由图 12 可见, semSensFuzz 误导角度的上四分位数比 IATG 在参数设置为 S1 和 S2 时分别小 8.75°和 1.04°, 表明 semSensFuzz 的误导能力低于 IATG. 这是由于 semSensFuzz 的改变物体颜色方法缺乏针对性, 只能对所有物体逐一变异, 与使用解释方法指导物体选择并将其替换为其他物体的 IATG 相比, 误导能力较弱. 如图 13 可见, semSensFuzz 所生成测试数据的 LPIPS 值的上四分位为 IATG 参数设置为 S1 和 S2 的 15.3% 和 156.0%, 这表明 IATG 在参数设置为 S2 时生成的测试数据相比 semSensFuzz 更加接近原始图像, 但在参数设置为 S1 时生成的测试数据相比 semSensFuzz 与原始数据的差距更大.

为更直观地展示不同方法生成测试数据的真实性差异, 图 14 展示了不同方法所生成部分测试数据的示例. 其中每一列为原始数据及不同生成方法所生成的测试数据 (DeepTest 仅展示了误导能力较强的 Rotation 和 Shear 两

种方法). 从图 14 中可以看出, IATG 在参数设置为 S1 和 S2 时均生成了较为真实的测试数据, 且所生成的测试数据没有改变物体语义和物体之间的位置关系, 因而具有和原始数据相同的测试预言. DeepTest 则对原始数据全局进行改动以生成测试数据, 虽然具有和原始数据相同的测试预言, 但其生成测试数据的变异方式难以出现在真实世界, 且全局性改动也使其具有较高的 LPSPS 值. semSensFuzz 改变物体颜色和添加物体两种方法只对单个物体进行变异且大多应用于尺寸较小的物体, 这使其生成的测试数据具有较小的 LPIPS 值, 但改变物体颜色的方法缺乏针对性, 会对所有物体逐一变异导致生成大量误导角度极低的测试数据. 而 semSensFuzz 添加物体的方法因添加新物体会改变测试数据的测试预言, 无法使用蜕变测试技术检测误导角度. 此外, semSensFuzz 生成的部分测试数据中所添加新物体位于其他物体之上, 虽然具有较低的 LPIPS 值但明显不符合常识. 例如第 7 行的第 2 张图片中 semSensFuzz 将一辆橙色汽车添加至行人前方且悬浮于地面上方.

图 14 不同方法生成测试数据示例

与 DeepTest 相比, IATG 明显降低了 LPIPS 值, 且转向角误导角度仅略低于 DeepTest. 与 semSensFuzz 相比, IATG 具有更高的误导角度, 且同样具有较低的 LPIPS 值. 这也引出了一个新的问题: 当误导角度相同时, IATG 相比 DeepTest 和 semSensFuzz 所生成测试数据的 LPIPS 值如何?

为此, 我们从 DeepTest 生成的 17 850 张测试数据和 semSensFuzz 生成的 50 000 张测试数据中分别随机采样出 5 408 个样本, 使其与参数设置为 S1 和 S2 时 IATG 所生成的测试数据量之和相同, 以比较 IATG、DeepTest 和 semSensFuzz 所生成测试数据的误导角度及对应 LPIPS 值. 图 15 为所生成的散点图, 从最大值看, 当参数设置为 S1 和 S2 时, IATG 所生成测试数据的最大 LPIPS 值分别为 0.173 和 0.087, DeepTest 的最大 LPIPS 值为 0.38, semSensFuzz 的最大 LPIPS 值为 0.056; 从分布区域看, IATG 所生成测试数据主要集中于 LPIPS 值小于 0.1 的区域, 而 DeepTest 主要集中于 LPIPS 值大于 0.1 的区域, semSensFuzz 则主要集中于 LPIPS 值小于 0.05 且误导角度

小于 20°的区域. 可以看出, 当误导角度相同时, DeepTest 所生成的测试数据与原始数据差距较大, 具有较高的 LPIPS 值, 而 IATG 和 semSensFuzz 则具有较低的 LPIPS 值.

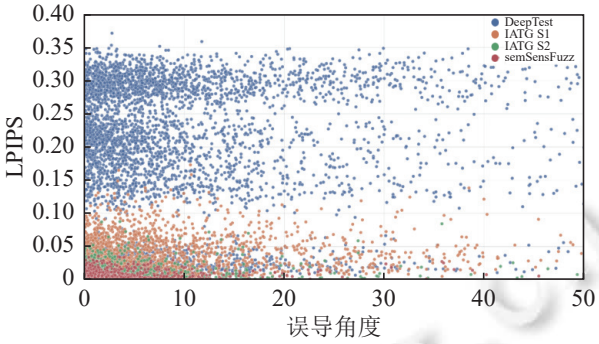


图 15 不同测试数据生成方法不同误导角度的 LPIPS 值

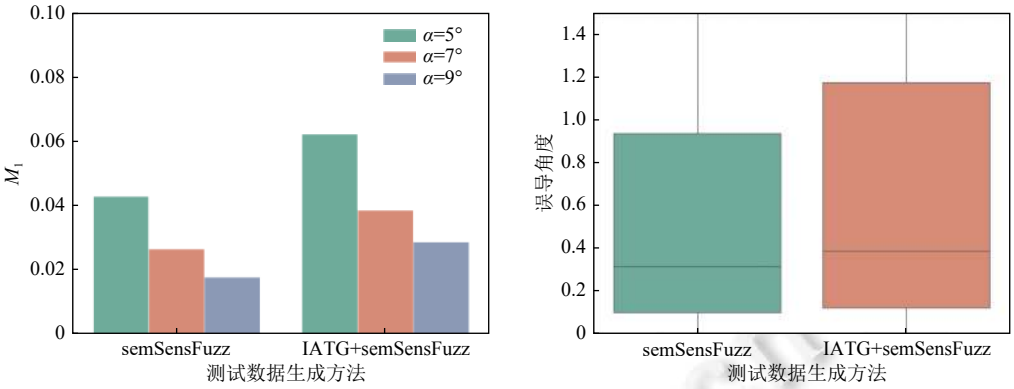
在 IATG 和 DeepTest 所生成测试数据中, 分别计算误导角度为 5º、15º、25º、35º和 45º (±1º) 测试数据的平均 LPIPS 值, 结果如表 1 所示. 表 1 中的括号外的数值表示不同方法所生成的测试数据中为相应误导角度时的平均 LPIPS 值, 括号内的数值为不同方法所生成的测试数据为相应误导角度的比例. 例如: 第 1 行第 1 列的 0.193 8 (10.02%) 表示 DeepTest 所生成测试数据中有 10.02% 误导角度为 (5±1)º, 且平均 LPIPS 值为 0.1938. 在表 1 中, 当 IATG 参数设置为 S2 时, 虽然没有生成误导角度恰好为 45º±1º的测试数据, 但其所生成测试数据的最大误导角度可达 76.2º. 从表 1 可以看出, 当参数设置为 S1 和 S2 时, IATG 所生成测试数据的误导成功率略低于 DeepTest, 但相同误导角度的平均 LPIPS 值不到 DeepTest 的 22.5%. 这表明在误导角度相同时, IATG 所生成测试数据比 DeepTest 更加接近真实图像. 尽管 IATG 在相同误导角度的平均 LPIPS 值高于 semSensFuzz, 但其误导成功率远高于 semSensFuzz, IATG 在参数设置为 S1 和 S2 时误导成功率分别为 semSensFuzz 的 4.8–37 倍和 1.9–7 倍.

表 1 不同方法生成相同误导角度测试数据的平均 LPIPS 值

测试数据生成方法	误导角度				
	5º±1º	15º±1º	25º±1º	35º±1º	45º±1º
DeepTest	0.193 8 (10.02%)	0.199 5 (2.78%)	0.198 0 (1.15%)	0.205 7 (0.61%)	0.210 4 (0.40%)
semSensFuzz	0.004 9 (2.41%)	0.007 8 (0.16%)	0.011 1 (0.05%)	0.012 5 (0.01%)	0.005 4 (0.03%)
IATG (S1)	0.034 7 (11.47%)	0.044 8 (2.51%)	0.031 5 (0.97%)	0.032 2 (0.36%)	0.041 8 (0.16%)
IATG (S2)	0.009 5 (4.53%)	0.013 0 (0.78%)	0.016 0 (0.10%)	0.043 3 (0.07%)	—

semSensFuzz 改变物体颜色方法可在较低的 LPIPS 值下达到较高的误导角度, 但缺乏有针对性的物体选择方法使其误导成功率很低. 为探究 IATG 基于解释分析的重要物体选择方法是否能提高 semSensFuzz 的误导能力, 我们使用 IATG 基于解释分析的方法帮助 semSensFuzz 选择重要物体, 再改变颜色以生成测试数据, 其中重要物体选择方法的参数设置与 S2 相同. 将此方法生成的 6 690 张测试数据与原 semSensFuzz 方法所生成测试数据的 M_1 指标及误导角度进行了对比, 实验结果如图 16 所示. 在引入 IATG 基于解释分析的重要物体选择方法后, 对于不同误导角度, semSensFuzz 的 M_1 指标提高了 1.45 倍至 1.61 倍, 总体误导角度的平均值提高了 1.29 倍. 这表明 IATG 的重要物体选择方法有效提高了 semSensFuzz 的转向角误导能力.

针对 RQ3 的结论: 与 DeepTest 相比, 在误导角度相同时, IATG 所生成测试数据更接近原始真实图像, LPIPS 值不到 DeepTest 的 22.5%. 与 semSensFuzz 相比, 在相同误导角度时, 尽管 IATG 生成测试数据 的平均 LPIPS 值更高, 但总体误导角度的平均值和误导成功率分别比 semSensFuzz 高至少 4.8 倍和 1.9 倍. 此外, IATG 的重要物体选择方法可有效提高 semSensFuzz 生成测试数据的误导能力.



(a) M_1 (b) 总体误导角度

图 16 semSensFuzz 在引入 IATG 重要物体选择方法前后的误导能力对比

4 有效性分析

本节从内部有效性、外部有效性和构造有效性 3 个方面对所提方法进行有效性分析.

本方法的内部有效性影响主要在于两个方面. 首先是测试数据的生成过程是否正确, IATG 对 CNN 模型的解释方法使用 Grad-CAM 的第三方开源实现代码, 且实验中编写的物体选择及替换代码经过了多次内部审查, 以尽量确保代码实现的正确性. 其次是实验使用的评价指标 LPIPS[31]实现是否正确, 实验中 LPIPS 评价指标的实现代码和所使用的预训练 CNN 模型均来自开源的第三方项目 IQA-optimization[37], 以尽量确保所计算 LPIPS 值的正确性.

本方法的外部有效性影响主要在于 3 个方面. 一是实验的目标模型和所使用的数据集是否具有代表性, 英伟达 DAVE[13]自动驾驶架构被广泛用于构建转向角预测模型[5−7], 实验对象为使用该架构的预训练模型, 有一定的代表性, 但无法保证 IATG 在其他架构或预训练模型上的有效性. 实验中真实物体提取和图像翻译方法分别使用公开数据集 Cityscapes[14]和 BDD100K[15], Cityscapes 还作为了测试数据生成的原始图像来源. 其在自动驾驶和计算机视觉领域被广泛使用[9,30]. 以上数据集具有一定的代表性, 但无法保证在其他数据集上是否能得到相同的结果. 二是实验结果是否具有代表性, IATG 不存在随机性因素, 而作为比较对象的 $IATG_{rand}$ 具有随机性, 但箱型图去除了不符合正态分布的异常值, 一定程度上排除了随机因素对实验结论的影响. 三是将 IATG 应用于其他自动驾驶软件的可扩展性. 除转向角预测模型之外, IATG 可用于其他使用 CNN 的自动驾驶模型. 对于其他使用图像数据作为输入, 并具有自我可解释性或外部可解释性的非 CNN 自动驾驶软件, 需要将 Grad-CAM 替换为其他相应的解释方法以扩展 IATG. 本文实验部分只重点关注了使用 CNN 构建的转向角预测模型, 无法保证 IATG 在其他模型上的有效性.

本方法的构造有效性影响主要在于评价测试数据指标的有效性, 实验中使用误导转向角超过一定阈值的数据占总测试数据的比例来评价误导能力, 这一指标在自动驾驶决策模块的测试研究中被广泛使用[5−7]. 另外实验部分使用 LPIPS 值评价测试数据与原始图像的接近程度, 其作为计算机视觉领域的主流评价指标, 被广泛用做评价图像的相似程度[38]、合成视频帧的质量[39]和生成图像与真实图像的接近程度[40]等指标.

5 相关工作

本节将介绍与本文相关的解释方法和其他自动驾驶软件的缺陷检测方法, 并与 IATG 进行对比.

5.1 深度学习软件的解释方法

为了提高深度学习软件的可解释性和透明性, 学术界对深度学习软件的解释方法进行了广泛和深入的研究并且提出了一系列方法.

以 CNN 为例, Zhou 等人[25]提出了基于类激活映射的 CNN 解释方法 CAM, 利用全局平均池化 (global average pooling) 层来替代 CNN 模型中除 Softmax 层以外的所有全连接层, 并通过将输出层的权重投影到卷积特征图来识别图像中的重要区域. 具体地, CAM 首先利用全局平均池化操作输出 CNN 最后一个卷积层每个特征图的空间平均值, 并通过对空间平均值进行加权求和得到 CNN 的最终决策结果. 同时, CAM 通过计算最后一个卷积层的特征图的加权和, 得到 CNN 模型的类激活图, 其反映了 CNN 用来识别该类别的重要图像区域. 最后, 通过热图的形式可视化类激活图得到视觉解释结果.

然而, CAM 方法需要修改网络结构并重训练模型, 因而实用性不佳. 为此, Selvaraju 等人[26]对 CAM 方法进行了改进, 提出了一种将梯度信息与特征映射相结合的梯度加权类激活映射方法 Grad-CAM. 给定一个输入数据, Grad-CAM 首先计算预测结果相对于最后一个卷积层中每一个特征图的梯度并对梯度进行全局平均池化, 以获得每个特征图的重要性权重. 然后, 基于重要性权重计算特征图的加权激活, 以获得一个粗粒度的梯度加权类激活图, 用于定位输入数据中 CNN 决策起重要影响的区域. 与 CAM 相比, Grad-CAM 无需修改网络架构或重训练模型, 避免了因修改模型或重训练带来的精度损失, 可适用于不同任务和结构的 CNN 模型. 因其具有良好的解释效果和兼容性, 本文选择将 Grad-CAM 作为 IATG 的解释方法.

5.2 自动驾驶软件的缺陷检测方法

致错场景 (fault scenario) 表示能够使待测系统发生错误的场景, 在深度学习软件的通用测试研究中, 测试数据生成方法被用于寻找致错场景. 在基于图片的致错场景生成方法中, 蜕变测试方法被广泛应用[41−43]. 为将基于致错场景的缺陷检测方法应用于深度学习目标检测系统, Wang 等人[44]提出了针对目标检测系统的蜕变测试方法 MetaOD, 该方法通过在背景图像中插入物体实例来合成接近真实的图像, 并使用蜕变关系在排除插入对象对识别结果的影响后, 判断原始图像和合成图像对于目标检测系统的等价性. MetaOD 工作流程包含物体实例的提取、选择和插入. MetaOD 对 TensorFlow API 提供的 4 个商业目标检测系统服务和 4 个预训练模型进行了测试. 实验结果表明, MetaOD 发现了数以万计的错误检测样本. 为了进一步展示 MetaOD 的实际应用, 他们使用检测出错误的合成图像对模型进行重新训练. 重训练模型的性能显著提高, MAP (mean average precision) 从 9.3% 提高到 10.5%. 由于缺少测试预言 (test oracle), MetaOD 生成的测试数据无法用于检测转向角预测模型缺陷. 受 MetaOD 启发, IATG 使用解释分析技术获取原图像中重要物体, 在不改变场景的前提下, 将其替换为语义相同的其他物体来生成测试数据, 并使用蜕变关系来检测转向角预测模型的缺陷.

在针对自动驾驶软件的测试工作中, 能够使待测模块发生错误的致错场景常被用于检测缺陷, 目前许多学者致力于研究快速高效生成致错场景的方法, 进而使自动驾驶软件发生错误行为. Tian 等人[6]以同一场景在不同的天气、背景下, 自动驾驶软件应当采取一致的行为作为基本原则, 提出 DeepTest 使用驾驶员在真实世界驾驶汽车采集的图片作为原始图像, 利用多种变换对其来生成测试数据. 其中包括, 线性变化 (即使用亮度和对比度两种途径调整图像, 以模拟遭遇强光、镜头失真等情况)、仿射变换 (即使用平移、缩放、水平剪切和旋转, 以模拟物体的位置变化、特殊的拍摄角度和拍摄异常等情况)、卷积变换 (即向图像加入雾气和雨天的效果, 用于模拟在特殊天气下各种交通情况). 实验结果表明 DeepTest 能有效提升生成图片对 DNN 的神经元覆盖率, 从而检验 DNN 模型的安全性和测试的充分性. 虽然真实世界可能因为图像采集设备的缺陷而使输入图像产生类似所使用的线性变化和仿射变换的情况, 但是在某种程度上也应该归咎于图像采集设备的缺陷导致, 不能完全认为是自动驾驶软件的缺陷. 与 IATG 相比, DeepTest 生成的测试数据和真实图像差异较大.

由于 DeepTest 的变化过于简单, 生成的图片较为粗糙且与真实图像差异较大, Zhang 等人[7]提出了 DeepRoad 以提高生成图片的质量, 他们使用了一种基于对抗生成网络的方法生成测试数据. DeepRoad 方法使用无监督图像转换模型, 将两个目标语义场景 (如晴天图像和雪天图像) 投影到相同的驾驶场景, 并经过训练将晴天图像合成为雪天图像. 实验结果表明, DeepRoad 生成的图片更接近于实际驾驶过程中采集到的视频和图片, 比 DeepTest 生成的图像更接近真实场景. 但 DeepRoad 生成的测试数据并非难以从现实世界采集, 只需要将装有图像和转向角采集设备的车辆由驾驶员在不同天气条件下, 在现实道路上驾驶便能采集到大量真实带转向角标签的测试数据. 而

在 IATG 所生成的场景相同的不同图像则难以进行采集.

在真实场景中捕捉的测试数据通常不会存在全局性的误导, 即真实场景中的误导因素通常是由图像中的某个物体或者某部分所引起的. 为了模拟真实世界的这一特性, Zhou 等人[5]提出了自动驾驶的系统物理世界测试方法 DeepBillboard, 他们选取了道路中常见的公告牌进行像素级别的对抗扰动. 公告牌的位置和区域会随着视角和距离发生变化, 他们在扰动过程中确保扰动只会出现在公告牌区域内. 实验结果表明, 这类基于公告牌扰动的对抗样本确实会对转向角预测模型产生误导. Kong 等人[45]提出的 PhysGAN 使用与 DeepBillboard 类似的方法生成对抗广告牌, 并放置在真实世界以误导自动驾驶软件的转向角预测模型. 不同之处在于 PhysGAN 生成的对抗广告牌是基于真实广告牌添加扰动像素生成的, 而 DeepBillboard 则是没有实际意义的像素色块. Kong 等人使用麦当劳和苹果公司的真实广告牌作为原始输入, 将添加对抗像素后的对抗广告牌放置在真实世界后经实验证明能有效误导自动驾驶软件的转向角预测模型, 并在外观上非常接近原始广告牌. DeepBillboard 和 PhysGAN 与 IATG 一样关注到测试数据中某个物体对自动驾驶软件的误导. 但与 IATG 不同, DeepBillboard 和 PhysGAN 所使用的是通过像素级扰动来生成对抗样本的方法, 所添加的扰动通常不会出现在真实世界.

以上的方法往往只注重发现错误行为, 而没有关注环境条件对自动驾驶软件的影响. Li 等人[36]提出的 TACTIC 方法, 可在不同的环境条件下测试基于深度学习的自动驾驶软件, 以识别关键的环境条件, 即自动驾驶软件更容易出错的环境条件. 针对环境条件空间过大的问题, 他们提出了一种基于搜索的方法来识别由图像翻译方法生成的关键环境条件. 大规模实验表明 TACTIC 与其他方法相比能够有效识别关键环境条件, 生成真实的测试图像, 同时揭示更多的错误行为. TACTIC 与 IATG 一样关注于寻找导致自动驾驶软件缺陷的因素, 而非盲目的生成大量测试数据. 不同之处在于 TACTIC 关注的是特定环境条件对自动驾驶软件的影响, 而 IATG 则关注相同场景的不同图像对自动驾驶软件的影响.

6 总结和未来工作

为检测自动驾驶软件中的缺陷, 本文提出的测试数据生成方法 IATG 利用深度学习的解释方法定位图像中的重要物体, 并对其进行替换来生成测试数据, 对自动驾驶软件的决策模块产生误导, 以检测其中的缺陷. 与其他测试数据生成方法相比, IATG 能有效检测基于 CNN 的转向角预测模型在面对场景相同的不同图像时产生不一致行为所导致的错误. 实验结果表明, IATG 生成的测数据能有效误导转向角预测模型做出错误决策, 且在误导角度相同时比 DeepTest 更加接近真实图像, 比 semSensFuzz 具有更高的误导能力, 且 IATG 基于解释分析的重要物体选择方法可有效提高 semSensFuzz 生成测试数据的误导能力.

本文是将解释方法应用于自动驾驶软件测试的初步探索. 未来, 我们将进一步研究 IATG 对自动驾驶软件其他模块进行测试的可行性.

References:

[1] Yu K, Jia L, Chen YQ, Xu W. Deep learning: Yesterday, today, and tomorrow. Journal of Computer Research and Development, 2013, 50(9): 1799–1804 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2013.20131180]

[2] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015, 521(7553): 436–444. [doi: 10.1038/nature14539]

[3] Chen CY, Seff A, Kornhauser A, Xiao JX. DeepDriving: Learning affordance for direct perception in autonomous driving. In: Proc. of the 2015 IEEE Int'l Conf. on Computer Vision. Santiago: IEEE, 2015. 2722–2730. [doi: 10.1109/ICCV.2015.312]

[4] Archyde. Tesla hits a white truck again with two passengers rushing to ICU. 2021. https://www. archyde. com/tesla-hits-a-white-truck-again-with-two-passengers-rushing-to-icu-tesla-tesla-electric-car

[5] Zhou HS, Li W, Kong ZL, Guo JF, Zhang YQ, Yu B, Zhang LM, Liu C. DeepBillboard: Systematic physical-world testing of autonomous driving systems. In: Proc. of the 42nd Int'l Conf. on Software Engineering. Seoul: IEEE, 2020. 347–358.

[6] Tian YC, Pei KX, Jana S, Ray B. DeepTest: Automated testing of deep-neural-network-driven autonomous cars. In: Proc. of the 40th Int'l Conf. on Software Engineering. Gothenburg: IEEE, 2018. 303–314. [doi: 10.1145/3180155.3180220]

[7] Zhang MS, Zhang YQ, Zhang LM, Liu C, Khurshid S. DeepRoad: GAN-based metamorphic testing and input validation framework for autonomous driving systems. In: Proc. of the 33rd IEEE/ACM Int'l Conf. on Automated Software Engineering. Montpellier: IEEE, 2018.

132–142. [doi: 10.1145/3238147.3238187]

[8] Csurka G, Perronnin F. An efficient approach to semantic segmentation. Int'l Journal of Computer Vision, 2011, 95(2): 198–212. [doi: 10.1007/s11263-010-0344-8]

[9] Woodlief T, Elbaum S, Sullivan K. Semantic image fuzzing of AI perception systems. In: Proc. of the 44th Int'l Conf. on Software Engineering. Pittsburgh: IEEE, 2022. 1958–1969. [doi: 10.1145/3510003.3510212]

[10] Zhu XL, Wang HC, You HM, Zhang WH, Zhang YY, Liu S, Chen JJ, Wang Z, Li KQ. Survey on testing of intelligent systems in autonomous vehicles. Ruan Jian Xue Bao/Journal of Software, 2021, 32(7): 2056–2077 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6266.htm [doi: 10.13328/j.cnki.jos.006266]

[11] Ijaz N, Wang YH. Automatic steering angle and direction prediction for autonomous driving using deep learning. In: Proc. of the 2021 Int'l Symp. on Computer Science and Intelligent Controls. Rome: IEEE, 2021. 280–283. [doi: 10.1109/ISCSIC54682.2021.00058]

[12] Udacity. Challenge #2: Using deep learning to predict steering angles. 2016. https://medium.com/udacity/challenge-2-using-deep-learning-to-predict-steering-angles-f42004a36ff3

[13] Bojarski M, Del Testa D, Dworakowski D, Firner B, Flepp B, Goyal P, Jackel LD, Monfort M, Muller U, Zhang JK, Zhang X, Zhao J, Zieba K. End to end learning for self-driving cars. arXiv:1604.07316, 2016.

[14] Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, Franke U, Roth S, Schiele B. The Cityscapes dataset for semantic urban scene understanding. In: Proc. of the 2016 IEEE Conf. on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 3213–3223. [doi: 10.1109/CVPR.2016.350]

[15] Yu F, Chen HF, Wang X, Xian WQ, Chen YY, Liu FC, Madhavan V, Darrell T. BDD100K: A diverse driving dataset for heterogeneous multitask learning. In: Proc. of the 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 2633–2642. [doi: 10.1109/CVPR42600.2020.00271]

[16] Menzel T, Bagschik G, Maurer M. Scenarios for development, test and validation of automated vehicles. In: Proc. of the 2018 IEEE Intelligent Vehicles Symp. Changshu: IEEE, 2018. 1821–1827. [doi: 10.1109/IVS.2018.8500406]

[17] The Guardian. Tesla driver dies in first fatal crash while using autopilot mode. 2016. https://www.theguardian.com/technology/2016/jun/30/tesla-autopilot-death-self-driving-car-elon-musk

[18] Sun Sentinel. Tesla crash: Officials likely to probe if autopilot driving system played role in most recent fatality. 2019. https://www.sun-sentinel.com/news/florida/fl-ne-ap-tesla-second-fed-agency-20190303-story.html

[19] Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In: Proc. of the 2012 IEEE Conf. on Computer Vision & Pattern Recognition. Providence: IEEE, 2012. 3354–3361. [doi: 10.1109/CVPR.2012.6248074]

[20] Ghenescu V, Mihaescu RE, Carata SV, Ghenescu MT, Barnoviciu E, Chindea M. Face detection and recognition based on general purpose DNN object detector. In: Proc. of the 2018 Int'l Symp. on Electronics and Telecommunications. Piscataway: IEEE. 2018. 1–4. [doi: 10.1109/ISETC.2018.8583861]

[21] Tobiyama S, Yamaguchi Y, Shimada H, Ikuse T, Yagi T. Malware detection with deep neural network using process behavior. In: Proc. of the 40th Annual Computer Software and Applications Conf. Atlanta: IEEE, 2016. 577–582. [doi: 10.1109/COMPSAC.2016.151]

[22] Rajpurkar P, Irvin J, Zhu K, Yang B, Mehta H, Duan T, Ding D, Bagul A, Langlotz C, Shpanskaya K, Lungren MP, Ng AY. CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning. arXiv:1711.05225, 2017.

[23] Ji SL, Li JF, Du TY, Li B. Survey on techniques, applications and security of machine learning interpretability. Journal of Computer Research and Development, 2019, 56(10): 2071–2096 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2019.20190540]

[24] Du MN, Liu NH, Song QQ, Hu X. Towards explanation of DNN-based prediction with guided feature inversion. In: Proc. of the 24th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining. London: ACM, 2018. 1358–1367. [doi: 10.1145/3219819.3220099]

[25] Zhou BL, Khosla A, Lapedriza A, Oliva A, Torralba A. Learning deep features for discriminative localization. In: Proc. of the 2016 IEEE Conf. on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 2921–2929. [doi: 10.1109/CVPR.2016.319]

[26] Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-CAM: Visual explanations from deep networks via gradient-based localization. Int'l Journal of Computer Vision, 2020, 128(2): 336–359. [doi: 10.1007/s11263-019-01228-7]

[27] Pang YX, Lin JX, Qin T, Chen ZB. Image-to-image translation: Methods and applications. IEEE Trans. on Multimedia, 2022, 24: 3859–3881. [doi: 10.1109/TMM.2021.3109419]

[28] Isola P, Zhu JY, Zhou TH, Efros AA. Image-to-image translation with conditional adversarial networks. In: Proc. of the 2017 IEEE Conf. on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017. 5967–5976. [doi: 10.1109/CVPR.2017.632]

[29] Chen QF, Koltun V. Photographic image synthesis with cascaded refinement networks. In: Proc. of the 2017 IEEE Int'l Conf. on Computer Vision. Venice: IEEE, 2017. 1520–1529. [doi: 10.1109/ICCV.2017.168]

[30] Wang TC, Liu MY, Zhu JY, Tao A, Kautz J, Catanzaro B. High-resolution image synthesis and semantic manipulation with conditional

GANs. In: Proc. of the 2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 8798–8807. [doi: 10.1109/CVPR.2018.00917]

[31] Zhang R, Isola P, Efros AA, Shechtman E, Wang O. The unreasonable effectiveness of deep features as a perceptual metric. In: Proc. of the 2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 586–595. [doi: 10.1109/CVPR.2018.00068]

[32] Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: From error visibility to structural similarity. IEEE Trans. on Image Processing, 2004, 13(4): 600–612. [doi: 10.1109/TIP.2003.819861]

[33] Wang Z, Simoncelli EP, Bovik AC. Multiscale structural similarity for image quality assessment. In: Proc. of the 37th Asilomar Conf. on Signals, Systems & Computers. Pacific Grove: IEEE, 2003. 1398–1402. [doi: 10.1109/ACSSC.2003.1292216]

[34] Zhang L, Zhang L, Mou XQ, Zhang D. FSIM: A feature similarity index for image quality assessment. IEEE Trans. on Image Processing, 2011, 20(8): 2378–2386. [doi: 10.1109/TIP.2011.2109730]

[35] Yuan SL, Guo Z. A warning model for vehicle collision on account of the reaction time of the driver. Journal of Safety and Environment, 2021, 21(1): 270–276 (in Chinese with English abstract). [doi: 10.13637/j.issn.1009-6094.2019.0830]

[36] Li Z, Pan MX, Zhang T, Li XD. Testing DNN-based autonomous driving systems under critical environmental conditions. In: Proc. of the 38th Int'l Conf. on Machine Learning. PMLR, 2021. 6471–6482.

[37] Ding KY, Ma KD, Wang SQ, Simoncelli EP. Comparison of full-reference image quality models for optimization of image processing systems. Int'l Journal of Computer Vision, 2021, 129(4): 1258–1281. [doi: 10.1007/s11263-020-01419-7]

[38] Karras T, Laine S, Aila T. A style-based generator architecture for generative adversarial networks. In: Proc. of the 2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 4396–4405. [doi: 10.1109/CVPR.2019.00453]

[39] Chan C, Ginosar S, Zhou TH, Efros A. Everybody dance now. In: Proc. of the 2019 IEEE/CVF Int'l Conf. on Computer Vision. Seoul: IEEE, 2019: 5932–5941. [doi: 10.1109/ICCV.2019.00603]

[40] Karras T, Laine S, Aittala M, Hellsten J, Lehtinen J, Aila T. Analyzing and improving the image quality of StyleGAN. In: Proc. of the 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 8107–8116. [doi: 10.1109/CVPR42600.2020.00813]

[41] Pei KX, Cao YZ, Yang JF, Jana S. DeepXplore: Automated whitebox testing of deep learning systems. In: Proc. of the 26th Symp. on Operating Systems Principles. Shanghai: ACM, 2017. 1–18. [doi: 10.1145/3132747.3132785]

[42] Borkar TS, Karam LJ. DeepCorrect: Correcting DNN models against image distortions. IEEE Trans. on Image Processing, 2019, 28(12): 6022–6034. [doi: 10.1109/TIP.2019.2924172]

[43] Xie XF, Ma L, Juefei-Xu F, Xue MH, Chen HX, Liu Y, Zhao JJ, Li B, Yin JX, See S. DeepHunter: A coverage-guided fuzz testing framework for deep neural networks. In: Proc. of the 28th ACM SIGSOFT Int'l Symp. on Software Testing and Analysis. Beijing: ACM, 2019. 146–157. [doi: 10.1145/3293882.3330579]

[44] Wang S, Su ZD. Metamorphic object insertion for testing object detection systems. In: Proc. of the 35th IEEE/ACM Int'l Conf. on Automated Software Engineering. Melbourne: IEEE, 2020. 1053–1065.

[45] Kong ZL, Guo JF, Li A, Liu C. PhysGAN: Generating physical-world-resilient adversarial examples for autonomous driving. In: Proc. of the 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 14242–14251. [doi: 10.1109/CVPR42600.2020.01426]

附中文参考文献:

[1] 余凯, 贾磊, 陈雨强, 徐伟. 深度学习的昨天、今天和明天. 计算机研究与发展, 2013, 50(9): 1799–1804. [doi: 10.7544/issn1000-1239.2013.20131180]

[10] 朱向雷, 王海弛, 尤翰墨, 张蔚珩, 张颖异, 刘爽, 陈俊洁, 王赞, 李克秋. 自动驾驶智能系统测试研究综述. 软件学报, 2021, 32(7): 2056–2077. http://www.jos.org.cn/1000-9825/6266.htm [doi: 10.13328/j.cnki.jos.006266]

[23] 纪守领, 李进锋, 杜天宇, 李博. 机器学习模型可解释性方法、应用与安全研究综述. 计算机研究与发展, 2019, 56(10): 2071–2096. [doi: 10.7544/issn1000-1239.2019.20190540]

[35] 袁守利, 郭铮. 考虑驾驶员反应时间的车辆碰撞预警模型. 安全与环境学报, 2021, 21(1): 270–276. [doi: 10.13637/j.issn.1009-6094.2019.0830]

谢瑞麟(1996－), 男, 硕士生, CCF 学生会员, 主要研究领域为智能软件工程, 可信人工智能.

陈翔(1980－), 男, 博士, 副教授, CCF 高级会员, 主要研究领域为软件测试, 软件维护, 实证软件工程, 软件仓库挖掘.

崔展齐(1984－), 男, 博士, 教授, CCF 高级会员, 主要研究领域为智能软件工程, 可信人工智能.

郑丽伟(1979－), 男, 博士, 副教授, CCF 专业会员, 主要研究领域为需求工程, 群体协同, 大数据挖掘.