基于自动语义编辑的目标检测测试数据生成方法

陈皓明[1],桂智明[1],刘艳芳[2],范鑫鑫[3],路云峰[4]
(1. 北京工业大学 计算机学院,北京 100124;
2. 北京航空航天大学 计算机学院,北京 100083;
3. 中国科学院 计算技术研究所,北京 100190;
4. 北京航空航天大学 可靠性与系统工程学院,北京 100088)

摘 要:目标检测系统的测试数据生成对评估模型性能和发现潜在缺陷至关重要。现有方法在生成数据的多样性和真实性方面仍存在局限。该文提出了一种基于自动语义编辑的目标检测测试数据生成方法 SemaGen,通过构建高质量语义对象库并结合自动化语义编辑策略,实现对图像的插入、删除和替换等高级语义操作。具体而言,该方法首先通过多重筛选机制构建语义对象库,确保对象的语义完整性和场景适应性;其次,利用场景复杂度量化模型,综合考虑背景占比、实例数量和空间分布等因素,实现编辑策略的自适应选择;最后,提出基于对象重要性的替换策略、迭代式删除方法以及考虑语义相似度的智能插入机制,确保生成图像的真实性和多样性。实验结果表明,SemaGen 在三种对象操作任务上显著优于现有方法,生成的图像质量更高,FID 得分更优,证实了该方法在生成数据质量方面的优越性。在目标检测模型测试中,SemaGen 成功发现 YOLO v11、SSD 和 Mask R-CNN 等主流检测器在复杂场景下的性能缺陷,为目标检测测试用例生成提供了新的思路和工具。

关键词:目标检测;语义编辑;测试数据生成;深度神经网络;图像生成
中图分类号:TP391 **文献标识码**:A **文章编号**:1673-629X(2025)07-0016-08
doi:10.20165/j.cnki.ISSN1673-629X.2025.0052

Test Data Generation for Object Detection Based on Automated Semantic Editing

CHEN Hao-ming[1],GUI Zhi-ming[1],LIU Yan-fang[2],FAN Xin-xin[3],LU Yun-feng[4]
(1. School of Computer Science and Technology,Beijing University of Technology,Beijing 100124,China;
2. School of Computer Science and Engineering,Beihang University,Beijing 100083,China;
3. Institute of Computing Technology,Chinese Academy of Sciences,Beijing 100190,China;
4. School of Reliability and Systems Engineering,Beihang University,Beijing 100088,China)

Abstract:Test data generation for object detection systems is crucial for evaluating model performance and identifying potential defects. Existing methods still have limitations in generating diverse and realistic data. We present SemaGen,a test data generation method for object detection based on automated semantic editing,which achieves advanced semantic operations such as insertion,deletion,and replacement through constructing high-quality semantic object libraries and combining automated editing strategies. Specifically,the proposed method first constructs a semantic object library through multiple screening mechanisms to ensure object semantic integrity and scene adaptability. Secondly,it utilizes a scene complexity quantification model that comprehensively considers background ratio,instance quantity,and spatial distribution to achieve adaptive selection of editing strategies. Finally,it proposes an object importance-based replacement strategy,an iterative deletion method,and an intelligent insertion mechanism considering semantic similarity to ensure the authenticity and diversity of generated images. The experimental results show that SemaGen significantly outperforms the existing methods on the three object manipulation tasks,generates higher quality images with better FID scores,and confirms its superiority in generating data quality. In object detection model testing,SemaGen successfully identifies performance deficiencies of mainstream detectors such as YOLO v11,SSD,and Mask R-CNN in complex scenarios,providing new insights and tools for generating object detection test

收稿日期:2024-11-22 **修回日期**:2025-03-26
基金项目:复杂关键软件环境全国重点实验室自主课题(SKLSDE-2023ZX-17)
作者简介:陈皓明(1999-),男,硕士研究生,研究方向为智能软件测试;桂智明(1976-),男,博士,副教授,研究方向为机器学习;通信作者:路云峰(1980-),男,博士,副研究员,研究方向为群体智能系统安全与测试。

cases.

Key words: object detection; semantic editing; test data generation; deep neural networks; image generation

0　引　言

随着深度学习技术在目标检测领域的广泛应用,其在自动驾驶、智能监控和医疗影像分析等实际场景中展现了重要价值。然而,目标检测模型的复杂性及其“黑盒”特性使得模型在复杂场景下的鲁棒性评估面临巨大挑战。例如,在 2016 年特斯拉自动驾驶系统未能识别白色卡车的事故中[1]暴露了目标检测系统在处理极端场景时的不足。这些事件凸显了高质量测试数据对于全面评估目标检测系统性能的重要性。

目前用于测试深度神经网络的技术主要分为白盒测试和黑盒测试。白盒测试技术主要关注结构覆盖率(例如路径覆盖[2])和功能覆盖率,但需要对目标模型有充分了解,这限制了其应用场景。而黑盒测试技术则不需要对目标模型的先验知识,主要通过查询来发现引起错误的输入,或是蜕变测试来解决测试预言(oracle)问题,例如 IATG[3]提出的基于解释分析的自动驾驶软件测试方法。在测试数据生成方面,MetaOD[4]首次提出通过对象插入的方式生成测试数据,但由于不能理解图像语义,其只能插入图像中原有对象,导致生成数据的多样性不足。随后也有研究通过对象移除来构建自动驾驶的测试数据[5],但其仅使用噪声遮盖对象的方式难以保证生成图像的真实性。最近的 ObjTest[6]系统地提出了包含对象替换、删除和添加的数据生成方法,但其在处理复杂场景时仍存在生成内容真实性不足的问题。

虽然语义编辑搭配扩散模型可以在保证图像真实性的同时提供多样化的内容,但传统的语义编辑需要大量人工参与,难以满足自动化测试的需求。针对现有方法的局限性,该文提出了一种基于自动语义编辑的测试数据生成方法—SemaGen。该方法通过算法自动地对图像中的对象进行语义层面的编辑操作,并利用扩散模型生成测试用例。具体而言,该方法首先构建高质量语义对象库,通过多重筛选机制确保对象的语义完整性和场景适应性;其次,通过场景复杂度量化模型,综合考虑背景占比、实例数量和空间分布等因素,实现编辑策略的自适应选择;最后,在对象操作方面,提出基于对象重要性的替换策略、迭代式删除方法以及考虑语义相似度的智能插入机制,并设计保持类别平均尺寸的对象调整算法,从而确保生成图像的真实性和多样性。

本研究的主要贡献包括:

(1)提出了一种新颖的自动语义编辑技术,该技术结合对象添加、移除和替换操作,并通过场景复杂度量化模型自适应选择最优编辑策略;

(2)设计并实现了一种高效的测试数据生成方法,包括语义对象的提取、处理与编辑,利用扩散模型在保证生成图像真实性的同时,自动生成多样化的测试数据。

(3)通过对比实验验证了该方法的有效性,结果表明该方法不仅在生成数据的质量上优于现有方法,而且能够更有效地暴露目标检测系统的潜在缺陷。

1　研究背景

1.1　目标检测测试数据生成技术背景

随着深度学习技术的快速发展,目标检测已成为计算机视觉领域最具代表性的任务之一。从最初基于 Haar 特征和支持向量机(SVM)的传统方法,到现代深度学习方法中的两阶段检测器(如 R-CNN 系列[7])和单阶段检测器(如 YOLO[8]、SSD 系列[9]),目标检测技术在检测精度和效率上都取得了显著进展。然而,这些基于深度学习的目标检测系统虽然性能优越,但其“黑盒”特性也为系统测试带来了前所未有的挑战。

在深度学习模型中,系统行为主要由训练数据决定,而非传统意义上的代码逻辑。因此,为目标检测系统生成高质量的测试数据就显得尤为重要。目前,测试数据生成技术主要可以分为基于覆盖率驱动和基于蜕变关系两大类方法。基于覆盖率驱动的方法借鉴了软件测试中的模糊测试[10]思想,这些方法通过最大化神经元覆盖率等指标来生成测试数据[11-12]。然而,研究表明[13]这种方法存在明显局限性:不仅容易产生大量不真实的图像,而且更高的神经元覆盖率也并不能保证发现更多的缺陷。更重要的是,由于需要获取模型的内部结构信息,这类方法在实际应用中受到很大限制。

相比之下,基于蜕变关系的测试数据生成方法展现出更大的潜力[14-15]。该方法不依赖于模型内部结构,而是通过定义输入变化与输出变化之间的关系来指导数据生成。这种基于蜕变关系的方法具有显著优势:它不需要预先定义具体的测试预期结果,能够自动化地生成大量测试用例,最重要的是可以更好地保证生成数据的真实性和有效性。

然而,现有的基于蜕变关系的测试数据生成方法在应用于目标检测系统时仍面临诸多挑战。首先,生成数据的多样性往往不足,难以全面覆盖目标检测系统可能遇到的各种场景;其次,在处理复杂场景时难以保证图像质量,这直接影响了测试的有效性;这些问题

的存在凸显了在目标检测领域开发更有效的测试数据生成技术的重要性。特别是考虑到目标检测系统在自动驾驶、医疗诊断等关键领域的广泛应用,提高测试数据的质量和多样性,以及确保测试过程的有效性变得尤为重要[16]。这需要在保持蜕变关系方法优势的基础上,针对目标检测任务的特点,开发更有针对性的测试数据生成技术。

1.2 语义编辑技术背景

语义编辑技术最初在自然语言处理领域得到广泛应用,用于理解和操作文本的深层含义。这种技术能够在保持文本原意的同时,对其进行智能化的修改和重构。随着深度学习技术的发展,语义编辑的概念和方法逐渐被引入到计算机视觉领域,为图像生成和处理带来了新的可能性。

在计算机视觉领域,语义编辑技术已经在多个方面展现出其重要性和应用潜力。然而,在自动化测试目标识别系统的场景中,传统的人工语义编辑方法存在一些局限性。首先,人工编辑耗时耗力,难以满足大规模测试的需求。其次,人工编辑可能引入主观偏见,无法全面覆盖所有可能的语义变化。自动语义编辑技术则可以通过高效生成大量多样化的测试样本,自动探索目标识别系统的决策边界,发现潜在的错误和弱点。因此,自动语义编辑技术在测试目标识别系统中具有重要的创新意义。

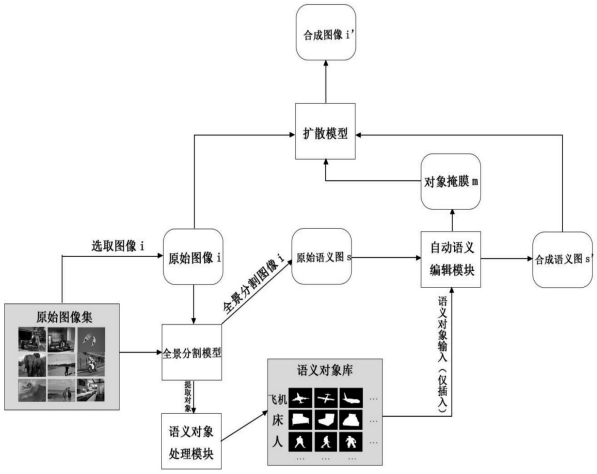

图 1 SemaGen 工作流程

2 方法与设计

为了生成用于自动化测试目标检测系统的合成图像,SemaGen 实现了一个自动化语义编辑的测试用例生成方法,图 1 展示了该方法的整体流程。该方法主要通过语义对象提取、语义对象处理、自动化语义编辑(包括对象插入、替换与删除)、扩散模型生成图像几个关键的步骤来生成合成图像。具体来说,给 SemaGen 提供一组图像,其通过全景分割模型执行全景分割来识别和提取对象实例。随后这些对象实例通过语义对象处理模块进行进一步的精炼与过滤,从而得到用于对象插入的语义对象库。随后,给定一张原始图像 i 作为原始测试用例,自动化语义编辑模块会根据指定的策略(详见 2.2 节)对图像的语义图 s 进行编辑,生成合成语义图 s' 以及被编辑对象的掩膜图 m。最后,利用扩散模型对合成语义图进行局部扩散,生成相应的合成图像 i'。

2.1 语义对象库构建

本节将详细介绍语义对象提取与处理的具体方法,阐述如何从原始数据集中提取有价值的语义对象,并通过一系列精细的处理步骤,构建一个适用于自动化语义编辑的高质量对象库。

2.1.1 语义对象提取

语义对象提取是构建对象库的第一步。在对象提取过程中,采用了全景分割技术。具体地,为了实现高效且精确的对象提取,采用了先进的 MaskDINO 全景分割算法[17]。

在本研究中,使用 COCO Stuff[18] 数据集作为基础数据源,通过 MaskDINO 进行全景分割。随后根据分割的结果将提取出来的语义对象掩膜按照其语义标签信息保存到本地。这些语义对象掩膜为后续的对象处理和语义编辑奠定了基础。

2.1.2 语义对象处理

在完成语义对象提取后,需要对提取的对象进行一系列处理,以确保对象库中的掩膜的质量。处理策略包括初步筛选、相似性计算和异常检测。图 2 展示了一些低质量掩膜的示例(类别为飞机)。具体来说,左上角的掩膜对象面积过小,右上角的掩膜对象不连贯,左下角的掩膜对象是不完整的,右下角的掩膜对象对于飞机来说是一个不自然的形状。

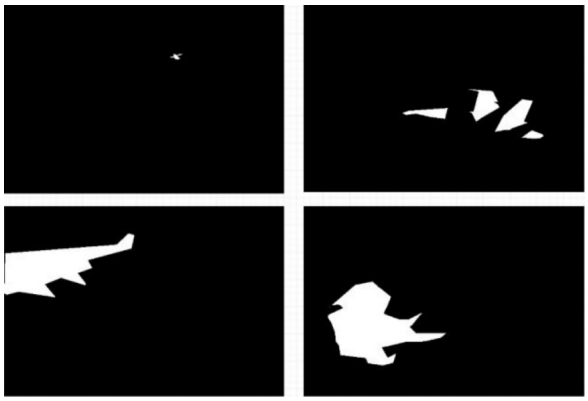

图 2 低质量掩膜示例

(1)掩膜初步筛选。

首先对从 COCO 数据集中提取的掩膜进行初步筛选,以确保只保留高质量的对象实例。筛选标准包括:(a)面积阈值:设置最小掩膜面积,过滤掉过小的

对象实例,这有助于去除噪声和不显著的对象。(b)连贯性检查:通过分析掩膜的连通区域,确保对象的完整性。如果某个连通区域的面积占总面积比例低于阈值,则认为该掩膜不完整,予以剔除。(c)边缘检查:排除靠近图像边缘的对象,以避免不完整或被截断的实例。设置了边缘阈值,任何触及这个边界的对象都会被过滤掉。

(2)相似度分析和异常检测。

为了评估对象库中掩膜的一致性,采用结构相似性指数(Structural Similarity Index,SSIM)进行相似度分析。具体来说,对每个掩膜,随机选择同类别的其他掩膜样本。计算目标掩膜与每个样本之间的 SSIM 值。取这些 SSIM 值的平均值作为该掩膜的相似性得分。

基于相似度分析的结果,使用隔离森林(Isolation Forest)算法进行异常检测,以识别和剔除不符合类别整体特征的掩膜。隔离森林算法的原理是:异常数据点更容易被隔离。该算法通过构建随机决策树来实现异常检测。检测完成后,根据检测结果,将被标记为异常的掩膜移除,并按照 SSIM 值从高到低选取 15 个最好的掩膜保存。

这种方法能有效识别那些可能是由于标注错误、对象遮挡或其他因素导致的异常样本。

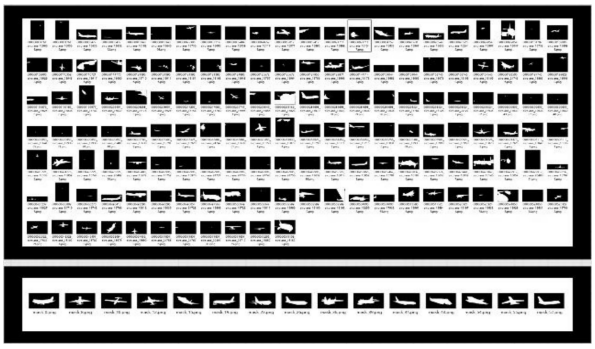

图 3　语义对象库示例

图 3 对比了处理前后的语义对象库,上部分是未处理之前的语义对象库,可以发现掩膜图质量参差不齐,尺寸不一。而下部分中的掩膜图质量高,尺寸统一。因此,通过语义对象处理模块的处理,能够有效地提高掩膜对象库的质量和一致性。这不仅确保了后续处理中使用的对象实例具有较高的代表性,还能缓解由于异常样本带来的潜在问题,从而提高生成图片的质量和对象插入的效率。

2.2　自动化语义编辑

本小节介绍提出的自动化语义编辑方法,该方法能够实现对图像中对象的替换、删除和插入操作。这些编辑操作在语义层面上进行,且不需要人工参与编辑的过程。在语义图中,一个对象的类别信息往往通过灰度值来标识,例如 person 对象的类别 id 为 1,则在语义图中像素值为 1 的区域表示这个实例对象为 person。接下来的对象替换、对象删除以及对象插入的操作的核心就是编辑语义图中的像素值。

2.2.1　基于内容分析的编辑策略

在自动化语义编辑过程中,关键步骤之一是通过深入分析图像内容来确定最优的编辑策略。该文提出了一种基于场景复杂度的编辑策略,让系统自动决定是否进行对象插入、删除和替换操作。算法 1 给出了自动化语义编辑的算法逻辑,如下所示:

Algorithm 1: 自动化语义编辑算法
Data:输入图像集 I
Result:合成图像集 S
1 初始化全景分割模型和扩散模型;
2 for 输入图像 i do
3　获取语义图和预测结果,计算复杂度 C;
4　if $C=0$ then
5　　跳过当前图像
6　end if
7　if $C \leqslant 0.3$ then
8　　执行对象插入和替换操作;
9　else
10　　执行对象插入、替换操作;
11　　while $C>0.3$ 且存在可移除对象 do
12　　执行对象移除操作,更新 C;
13　　end while
14　end if
15　扩散模型生成合成图像 i',加入 S;
16 end for

利用全景分割模型对原始图像 i 进行全景分割,可以得到图像中所有实例对象(thing)以及背景对象(stuff)的大小、位置以及类别信息,因此提出了一个数学模型来量化场景的复杂度。这个模型考虑了背景占比、实例数量、空间分布等多个因素。首先背景占比函数 B 定义如下:

$$B=\frac{A_{\text{background}}}{A_{\text{total}}} \tag{1}$$

其中,$A_{\text{background}}$ 表示背景区域像素面积,A_{total} 表示图像的总像素面积大小。另外定义空间分布复杂度函数 D 如下:

$$D=\frac{\sum A_{ij}}{M*(M-1)/2} \tag{2}$$

其中,A_{ij} 表示实例对象 i 和 j 之间是否相邻(若两个对象边界之间间隔 x 个像素以内,则认为对象是相邻的),如果两个实例对象在语义图中相邻,则 $A_{ij}=1$,否则 $A_{ij}=0$。M 是图中实例对象的总数。对于完全聚集的场景(所有实例相邻),D 等于 1;对于完全分散的场景(没有实例相邻),D 等于 0。D 值越小,表示实例在

空间上分布越分散。

有了以上的定义,可以定义场景复杂度函数 C 如下:

$$C = \alpha * (1 - B) + \beta * D \tag{3}$$

其中,权重系数 α ,β 满足 $\alpha + \beta = 1$。这个简化的数学模型专注于语义图的特性,提供了一个客观、可量化的方法来评估场景复杂度,从而指导后续的编辑策略决策。文中权重系数 α 取 0.8,β 取 0.2。

经过人工检查发现场景复杂度函数是贴合实际的,大部分图像的场景复杂度较低,只有小部分的图像场景复杂度较高。因此,制定以下策略:

(1)如果 $C > 0.6$,场景被认为高度复杂,系统倾向于选择删除或替换操作。

(2)如果 $0.3 \leqslant C \leqslant 0.6$,场景复杂度适中,此时系统会同时尝试删除、替换和添加操作。

(3)如果 $C < 0.3$,此时场景相对简单,系统会考虑插入和替换操作。

2.2.2 对象替换和删除

对象替换和删除是自动化语义编辑中的两个关键操作。这些操作基于前面描述的场景复杂度模型和编辑策略来执行。

(1)对象替换。

当系统决定进行对象替换时,它会遵循以下步骤:

(a)目标对象选取:基于场景复杂度分析,系统会优先选择对整体复杂度贡献较大的对象进行替换。这可能是占据较大面积的对象,也可能是和多个实例对象相邻的对象。

(b)替换类别选取:对象的替换可以分为同类别不同对象的替换以及不同类别不同对象的替换,不同类别的替换很可能会出现新类别的对象可能在大小、形状上与原对象的空间位置不匹配的情况,影响图像真实性。因此本研究中采用同类别替换,尽可能保证生成图像的真实性与合理性。

(c)替换执行:系统会根据替换对象的语义信息自动生成一张该对象的掩膜图 m ,并让扩散模型恢复原图中被该掩膜图覆盖的区域,即可生成对象替换后的图片 i' 。此操作不会编辑原始图像的语义图 s ,即通过自动语义编辑处理后,$s' = s$ 。

(2)对象删除。

对象删除操作通常在场景被认为过于复杂时执行,主要包括删除对象选择、删除操作执行以及删除策略三部分。具体步骤如下:

(a)删除对象选择:与对象替换相反,系统会优先选择那些对场景复杂度贡献较小的对象,或者是在空间上相邻对象较少的对象。需要注意的是,系统不会删除图像中唯一存在的对象,这样可以避免删除掉图像中的主体对象,从而造成图像真实性下降。

(b)删除操作执行:系统在确定删除对象后,首先会编辑原图的语义图,将该对象的像素值置为 0(背景的像素值为 0),同时会自动生成一张该对象的掩膜图 m ,然后利用扩散模型根据编辑后的语义图 s' ,原始图像 i 以及掩膜图 m 生成合成图像 i' 。

(c)迭代删除:在每次删除操作后,系统会重新计算场景复杂度,以决定是否可以进行进一步的删除操作,如果此时场景依旧复杂且图像中有一类别存在不是唯一的对象,系统会继续尝试对象删除直到场景复杂度 $C < 0.3$ 或图像中每一个类别都只剩下唯一的对象。

2.2.3 对象插入

对象插入操作主要在场景复杂度较低或适中时执行。这个过程涉及多个关键步骤,主要包括插入位置选取、插入对象选择以及语义对象插入。

(1)插入位置选取。

在对象插入过程中,系统首先从语义图中提取并分析背景区域。这些区域按面积大小降序排列,以优化潜在插入位置的选择。系统在符合条件的背景区域中随机选取插入位置,同时严格确保所选区域不与现有对象像素重叠,从而保持场景的完整性和语义一致性。

(2)插入对象选择。

该文提出了一种混合策略来优化插入对象的选择过程。首先,系统分析每个背景区域附近相邻的语义对象类别,建立局部语义环境理解。随后,利用 BERT 模型[19]计算相邻语义对象与语义库中各类别之间的词语相似度。最终,系统根据以下优先级序列确定待插入对象:(a)与相邻对象重合的类别;(b)相邻语义对象的类别;(c)具有较高词语相似度的类别。这种分层策略既保证了语义一致性,又提供了适度的对象多样性。

(3)语义对象插入。

在进行对象插入之前,系统采用“类别平均尺寸调整策略”来确定待插入对象的尺寸。具体而言,该策略分为两种情况:(a)对于在原图中已存在的类别,使用该类别现有对象的平均尺寸作为参考;(b)对于新类别对象,则采用原图中所有对象的平均尺寸。通过这种策略,系统能够在对象插入过程中,依据对象类别的存在性,动态调整待插入对象的尺寸,从而增强图像的整体视觉一致性。

2.2.4 扩散模型生成图像

在得到自动化语义编辑模块生成的语义图后,需要利用图像生成模型根据语义图进行图像生成。扩散模型是一种近年来在生成模型领域中崭露头角的深度

学习框架。与传统生成对抗网络(GANs)相比,扩散模型不仅在生成质量和多样性上展现出优越的性能,而且具有更稳定的训练过程和更强的可控性,能够更好地捕捉数据分布的细节特征,为图像生成任务提供了更可靠的解决方案。

该文采用 FreestyleNet[20] 模型实现从语义布局到真实图像的生成。FreestyleNet 是一种基于大规模预训练文本到图像扩散模型的方法,能够灵活地根据给定的语义布局和文本描述生成高质量图像。具体来说,模型接受原始图像 i,经过自动语义编辑的语义图 s' 以及编辑操作的对象掩膜图 m 作为模型输入,模型将生成相应的合成图像 i'。需要注意的是,仅仅只对掩膜 m 所表示的区域进行局部扩散生成,而保留原图中其他区域的细节不变。

3　实验与分析

3.1　实验设置

具体实验环境配置如表 1 所示。

表 1　实验环境配置

配置项	环境
编程语言	Python
深度学习框架	Pytorch1.10.2
操作系统	Ubuntu20.04
CPU	Intel(R) Xeon(R) W-2223
内存	32G
GPU	NVIDIA GeForce RTX 2080 Ti
CUDA	11.3

本研究的重点在于通用目标检测模型的测试数据生成,因此选取通用目标检测系统测试方法 MetaOD[4],MethodX[5](论文中没有给出具体方法名,以此名称作为代替)及 ObjTest[6] 作为基线,采用被广泛使用的 COCO Stuff 数据集作为原始图像集。数据集总共包含了 172 种类别,图像数量超过 164K,其中包含了 80 种对象类别(thing)以及 91 种背景环境类别(stuff)和 1 个未标记类(unlabeled)。具体来说,从验证集中选取了 2 000 张图片作为原始输入。目标检测模型选取常见的一阶段模型 YOLO v11、SSD 以及二阶段模型 Mask R-CNN。

为了全面评估不同测试方法在图像生成质量方面的表现,采用 FID(Fréchet Inception Distance)作为图像自然性的主要评估指标。FID 是衡量生成图像与真实图像之间分布差异的重要指标,其数值越低,表示生成图像的质量越高,与真实图像的视觉特征越接近。值得注意的是,针对 SemaGen 和 ObjTest 两种都存在对象替换、移除、添加操作的测试方法,对三种操作生成的图像分别评估 FID 值并取三者的平均值。

另外,本研究采用平均精度均值(mAP)作为目标检测模型的评估指标,其计算过程首先基于 IoU 值将检测结果分类为真正例(N_{TP})、假正例(N_{FP})和假负例(N_{FN}),然后计算精确度($N_{TP}/(N_{TP}+N_{FP})$)和召回率($N_{TP}/(N_{TP}+N_{FN})$),通过不同检测阈值绘制精确度-召回率曲线并计算曲线下面积得到每个类别的平均精度(AP),最后将所有类别的 AP 取平均得到 mAP 值。在本研究中,较低的 mAP 值表示模型对编辑后图像的检测结果与原始图像存在较大差异,意味着数据生成方法能够发现更多目标检测模型的错误,为后续的模型优化提供了重要依据。实验中统一设置 IoU 阈值和置信度阈值为 0.5,以确保评估的一致性。

3.2　实验结果

图 4 展示了不同测试数据生成方法在对象删除以及对象插入任务上的对比实验。在对象删除任务中,MethodX 通过简单地在目标区域添加噪声来扰乱模型对目标的检测,但这种方法生成的图像视觉效果较差,真实性较低。ObjTest 采用基于卷积神经网络的图像修复模型来移除目标对象,生成的图像质量优于 MethodX,但在边缘处理上存在不足,可能出现明显的模糊现象,甚至对其他对象的细节造成不必要的改动。相比之下,SemaGen 方法通过语义信息精准定位需要移除的对象,并利用扩散模型的注意力机制对目标区域进行精确控制,从而生成更真实的图像。SemaGen 在对象删除任务中的图像真实性和视觉一致性均优于 MethodX 和 ObjTest。

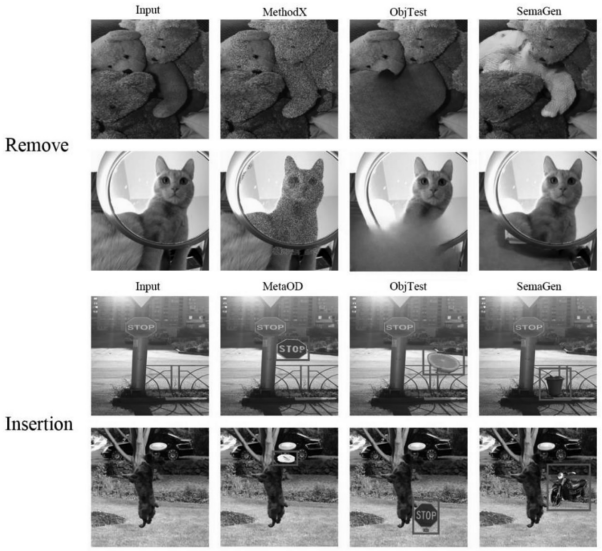

图 4　不同测试数据生成方法删除与插入操作对比

在对象插入任务中,MetaOD 和 ObjTest 均基于一个预定义的对象库,该对象库由现有的目标检测模型(如 MetaOD 中使用的 YOLACT)提取对象。然而,这

种方法存在两个主要问题:一是插入对象的多样性不足,二是对象库的质量受到目标检测模型性能的限制。相比之下,SemaGen 通过语义生成的方式插入对象,并采用了一系列优化策略以提升生成对象的质量,能够有效保证插入对象的多样性和真实性。在对象选择策略上,MetaOD 仅能插入原图中已有的对象,导致生成图像缺乏多样性;ObjTest 则不考虑原始图像的语义信息,随机插入对象,可能导致生成图像缺乏真实性,例如在马路上插入橘子或在草地上插入警示牌。SemaGen 通过对场景语义的深度理解,能够在合适的场景中插入多样化且合理的对象,例如在马路边插入垃圾桶,或在草地边插入摩托车,从而显著提升了生成图像的合理性和视觉一致性。

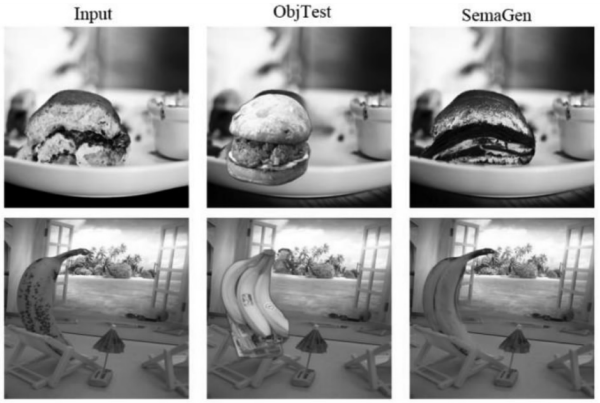

图 5 不同测试数据生成方法替换操作对比

图 5 展示了 ObjTest 和 SemaGen 两种测试数据生成方法在对象替换任务上的对比实验。在上部图像中,两种方法分别对汉堡进行替换操作;在下部图像中,则针对香蕉执行替换。ObjTest 采用的替换策略是基于两步操作:首先使用图像修复模型移除目标对象,随后执行对象插入操作。这种方法导致替换后的对象与周围环境存在明显的视觉不连贯性。相比之下,SemaGen 方法从语义层面出发,充分利用扩散模型对场景整体语义的理解能力,不仅提升了生成对象的质量,还显著增强了对象与环境的视觉一致性,从而在替换效果上明显优于 ObjTest 方法。

表 2 不同测试数据生成方法生成图像的得分

Method	FID
SemaGen	12.15
MetaOD	13.68
SemaGen(RI)	16.47
ObjTest	22.33
MethodX	31.63

表 2 展示了不同测试数据生成方法的 FID 得分对比。SemaGen 方法以 12.15 的最低 FID 值显著优于其他方法,其次是 MetaOD 的 13.68,虽然其 FID 值较低,但如图 4 所示,该方法生成的图像在多样性方面存在明显不足,往往表现出过度相似的特征。值得注意的是,即使采用随机插入策略(Random Insertion)的 SemaGen(RI),其 FID 值(16.47)仍优于 ObjTest(22.33)和 MethodX(31.63)。而标准 SemaGen 通过语义驱动的插入位置选择策略,相比随机插入策略降低了约 40% 的 FID 值,这一显著提升充分验证了插入位置选择策略的有效性,同时也凸显了其在生成任务中对场景语义理解和对象插入任务的优越性能。

表 3 三种目标检测模型在不同合成测试数据集上的平均精度均值

Model	mAP@50/%					
	Ori.	MetaOD	MethodX	ObjTest	SemaGen(RI)	SemaGen
YOLO v11	60.42	30.29	46.82	35.99	30.11	27.38
SSD	34.81	13.93	23.86	20.67	16.24	13.53
Mask R-CNN	55.28	30.26	40.46	33.24	31.96	27.79

表 3 展示了三种目标检测模型(YOLO v11、SSD 和 Mask R-CNN)在不同测试方法下的 mAP 值对比。从结果可以看出,标准 SemaGen 方法在降低目标检测性能方面表现最为突出。说明 SemaGen 通过其创新的语义编辑策略,能够更有效地模拟复杂场景中的干扰因素,从而更真实地评估模型的鲁棒性。此外,SemaGen 方法对不同类型的目标检测模型均表现出一致的干扰效果,说明其具有较强的通用性,能够适用于多种目标检测框架。通过显著降低模型的 mAP,SemaGen 方法能够更全面地暴露模型的潜在弱点,为后续的模型优化提供了重要依据。

4 结束语

该文提出了一种基于自动语义编辑的目标检测测试数据生成方法 SemaGen。该方法通过构建高质量的语义对象库并结合自动化语义编辑策略,实现了对图像的插入、删除和替换等语义级操作。实验结果表明,SemaGen 不仅在测试数据生成数量和多样性方面显著优于现有方法,还能生成更具代表性和挑战性的测试样本。特别是在针对 YOLO v11、SSD 和 Mask R-CNN 等主流检测器的测试中,生成的测试数据展现出较强的测试覆盖能力。这些成果为目标检测模型的性

能评估和测试用例生成提供了新的思路和工具。未来的研究方向将着重于进一步提升语义编辑的精确性和自然性,探索更多样化的编辑策略,以及扩展方法在更广泛测试场景下的适用性。同时,如何基于生成的测试数据来设计更全面的测试方案,提升目标检测模型的性能评估效率,也将是一个值得深入研究的方向。

参考文献:

[1] BOUDETTE N E. Tesla's self-driving system cleared in deadly crash[EB/OL]. 2017[2024-11-18]. https://nyti.ms/2iZ93SL.

[2] 钱忠胜,俞情媛,张　丁,等. 结合 SVM 与 XGBoost 的链式多路径覆盖测试用例生成[J]. 软件学报,2024,35(6):2795-2820.

[3] 谢瑞麟,崔展齐,陈　翔,等. IATG:基于解释分析的自动驾驶软件测试方法[J]. 软件学报,2024,35(6):2753-2774.

[4] WANG S,SU Z. Metamorphic object insertion for testing object detection systems[C]//2020 35th IEEE/ACM international conference on automated software engineering (ASE 2020). Virtual Event:IEEE,2020:1053-1065.

[5] WANG X,YANG S,SHAO J,et al. Object removal for testing object detection in autonomous vehicle systems[C]//2021 IEEE 21st international conference on software quality, reliability and security companion (QRS-C). Hainan:IEEE,2021:543-549.

[6] LIU Z X,FENG Y,XU J L,et al. ObjTest:object-level mutation for testing object detection systems[C]//15th Asia-Pacific symposium on internetware (internetware 2024). Macao:Association for Computing Machinery,2024:61-70.

[7] GIRSHICK R,DONAHUE J,DARRELL T,et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//2014 IEEE conference on computer vision and pattern recognition (CVPR). Columbus:IEEE,2014:580-587.

[8] REDMON J,DIVVALA S,GIRSHICK R,et al. You only look once:unified,real-time object detection[C]//2016 IEEE conference on computer vision and pattern recognition (CVPR). Seattle:IEEE,2016:779-788.

[9] LIU W,ANGUELOV D,ERHAN D,et al. SSD:single shot multibox detector[C]//Computer vision - ECCV 2016. Amsterdam:Springer,2016:21-37.

[10] GAN S,ZHANG C,QIN X,et al. CollAFL:path sensitive fuzzing[C]//2018 IEEE symposium on security and privacy (SP). San Francisco:IEEE,2018:679-696.

[11] PEI K,CAO Y,YANG J,et al. DeepXplore:automated whitebox testing of deep learning systems[C]//26th ACM symposium on operating systems principles (SOSP 2017). Shanghai:Association for Computing Machinery (ACM),2017:1-18.

[12] TIAN Y,PEI K,JANA S,et al. DeepTest:automated testing of deep-neural-network-driven autonomous cars[C]//40th ACM/IEEE international conference on software engineering (ICSE 2018). Gothenburg:Association for Computing Machinery (ACM),2018:303-314.

[13] HAREL-CANADA F,WANG L,GULZAR M A,et al. Is neuron coverage a meaningful measure for testing deep neural networks? [C]//28th ACM joint meeting on european software engineering conference and symposium on the foundations of software engineering (ESEC/FSE 2020). Virtual Event:Association for Computing Machinery (ACM),2020:851-862.

[14] 魏瑀皓,姚永明. 一种基于蜕变测试的卫星遥感目标检测模型鲁棒性测试方法[J]. 现代计算机,2023,29(20):35-39.

[15] 王　丹,王兴亚,黄　松,等. 基于蜕变测试的图像分类软件的鲁棒性评估方法[J]. 网络安全技术与应用,2023(12):41-44.

[16] 朱向雷,王海弛,尤翰墨,等. 自动驾驶智能系统测试研究综述[J]. 软件学报,2021,32(7):2056-2077.

[17] LI F,ZHANG H,XU H,et al. Mask DINO:towards a unified transformer-based framework for object detection and segmentation[C]//2023 IEEE/CVF conference on computer vision and pattern recognition (CVPR). Vancouver:IEEE,2023:3041-3050.

[18] CAESAR H,UIJLINGS J,FERRARI V. COCO-Stuff:thing and stuff classes in context[C]//2018 IEEE/CVF conference on computer vision and pattern recognition (CVPR). Salt Lake City:IEEE,2018:1209-1218.

[19] DEVLIN J,CHANG M W,LEE K,et al. BERT:pre-training of deep bidirectional transformers for language understanding[C]//2019 conference of the North American chapter of the association for computational linguistics:human language technologies (NAACL-HLT 2019). Minneapolis:Association for Computational Linguistics (ACL),2019:4171-4186.

[20] XUE H,HUANG Z W,SUN Q R,et al. Freestyle layout-to-image synthesis[C]//2023 IEEE/CVF conference on computer vision and pattern recognition (CVPR). Vancouver:IEEE,2023:14256-14266.