**显著物体检测项目文档**

一、选题背景介绍

1.1 研究背景与意义

显著物体检测（Saliency Object Detection, SOD）作为计算机视觉领域的核心技术之一，承担着模拟人类视觉注意力机制的关键任务。在实际应用场景中，其重要性尤为突出：在自动驾驶领域，SOD技术能够快速识别道路中的行人、车辆等关键目标，为决策系统提供可靠依据；在图像编辑场景下，可实现主体的自动分割与智能处理；而在医学影像分析中，更是辅助医生定位病变区域、提升诊断效率的重要手段。通过精准定位图像中的显著物体，SOD 技术不仅能够为后续的图像分析、目标识别等任务奠定基础，更能显著提升各类视觉系统的智能化水平。

1.2 研究现状

近年来，显著物体检测领域在学术界与工业界均取得了显著进展。早期基于手工特征的传统算法（如频域分析、区域对比度计算）因计算效率高而得到广泛应用，但面对复杂场景时泛化能力较弱。随着深度学习技术的崛起，基于全卷积网络（FCN）、编码器 - 解码器结构（如 U-Net、SegNet）以及注意力机制的算法逐渐成为主流，在检测精度和鲁棒性上实现了重大突破。然而，当前研究仍面临诸多挑战，如小目标检测精度不足、实时性与准确性难以平衡等问题，亟待进一步探索与优化。

1.3 选题动机

作为计算机视觉课程的实践项目，选择显著物体检测作为研究方向，旨在将课堂所学理论知识与实际应用相结合。本项目选取基于涂鸦标注的弱监督 SOD 算法（Scribble\_Saliency），既契合当前弱监督的研究热点，又能在有限的标注资源下实现高效检测，具有较强的创新性与实用价值。同时，通过项目实践，可系统掌握从算法设计、环境搭建到模型评估的全流程，深化对计算机视觉技术的理解与应用能力。

二、主流算法介绍

2.1 传统算法

1. 基于手工特征的方法：该类算法依赖人工设计的特征提取规则，如基于频域分析的谱残差法，通过计算图像在频域的能量分布差异来定位显著区域；基于区域对比度的算法则通过计算图像块间的颜色、纹理差异，突出与背景差异较大的区域。此类方法计算复杂度低，但特征表达能力有限，难以适应复杂场景。
2. 基于图模型的方法：将图像抽象为图结构，节点代表图像区域，边表示区域间的关系。典型算法如基于图割（Graph Cut）和随机游走（Random Walk）的方法，通过优化图的能量函数或计算节点间的转移概率，实现显著物体的分割。尽管这类方法在理论上能有效处理图像的全局信息，但计算效率较低，且对初始参数敏感。

2.2 深度学习算法

1. 全卷积网络（FCN）：通过将传统卷积神经网络中的全连接层替换为卷积层，FCN 实现了端到端的像素级预测。在 SOD 任务中，FCN 可直接输出与输入图像尺寸相同的显著图，大幅提升检测效率，但由于缺乏对上下文信息的有效利用，易出现边缘模糊等问题。
2. 编码器 - 解码器结构：以 U-Net、SegNet 为代表，这类网络采用对称的编解码架构，通过下采样提取高层语义特征，再通过上采样恢复空间分辨率，并结合跳跃连接融合多尺度信息。这种设计有效平衡了语义信息与空间细节，在显著物体检测中取得了优异表现。
3. 注意力机制：通过引入通道注意力（如 SENet）和空间注意力（如 CBAM）模块，模型能够自适应地聚焦于显著区域，抑制背景干扰。注意力机制的应用显著提升了模型对复杂场景和小目标的检测能力，成为当前 SOD 算法的重要改进方向。

三、选取算法

3.1 算法概述

本项目选取的Scribble\_Saliency 算法是一种基于弱监督学习的显著物体检测方案，其核心在于利用涂鸦标注（scribble annotations）替代传统的像素级标注，大幅降低数据标注成本。算法通过构建双分支网络，结合边缘特征提取与多尺度上下文信息融合，实现高效的显著区域预测，特别适用于标注资源有限的场景。

3.2 算法原理

* 网络结构：采用基于 VGG16 的Back\_VGG 模型，包含两个关键分支：一个分支用于提取图像的全局语义特征，另一个分支通过Edge\_Module和Atrous Spatial Pyramid Pooling（ASPP）模块增强边缘细节与多尺度特征表达能力。其中，Edge\_Module 通过多层卷积和上采样操作细化边缘信息，ASPP 模块则利用不同空洞率的卷积核捕获多尺度上下文。
* 损失函数：算法采用二元交叉熵损失（BCELoss）衡量预测显著图与真实标签的像素级差异，并引入平滑损失（smoothness\_loss）约束预测结果的空间连续性，通过联合优化两种损失函数，提升模型的鲁棒性与准确性。

3.3 算法优势

相较于其他算法，Scribble\_Saliency具有以下优势：

1. 弱监督特性：通过涂鸦标注即可训练模型，显著降低标注成本，适用于数据稀缺场景；
2. 结构轻量化：基于 VGG16 的改进架构在保证精度的同时减少计算量；
3. 多特征融合：Edge\_Module 与 ASPP 模块的结合有效提升了对复杂场景和小目标的检测能力。

四、环境搭建

4.1 硬件环境

|  |  |
| --- | --- |
| 设备类型 | 配置详情 |
| CPU | AMD Ryzen 5 5600H |
| GPU | NVIDIA GeForce RTX 3050Ti Laptop GPU（4GB 显存） |
| 内存 | 16GB |
| 存储 | 512GB SSD |

由于本地算力不足，项目训练阶段租用AutoDL GPU 云服务器，配置为：NVIDIA GeForce RTX 3090 GPU（24GB 显存），满足大规模数据训练需求。

4.2 软件环境

|  |  |
| --- | --- |
| 类别 | 版本信息 |
| 操作系统 | Ubuntu 20.04 |
| 编程语言 | Python 3.8 |
| 深度学习框架 | PyTorch 1.10.0 |
| 依赖库 | torchvision 0.12.0  scipy 1.7.1  pillow 8.3.2  pytorch 1.11.0  opencv-python 4.11.0.86  numpy 1.21.0 等 |

4.3 代码部署

1. 代码获取：通过 Git 命令克隆项目仓库：

git clone https://github.com/JingZhang617/Scribble\_Saliency.git

1. 数据集准备：
   * 下载S-DUTS 数据集（涂鸦标注版），解压后存放于/data/data目录；
   * 使用 Matlab 脚本generate\_gt\_mask\_from\_scribble.m将涂鸦标注转换为二值掩码，convert\_rgb2gray.m将 RGB 图像转为灰度图。
2. 模型训练：执行train.py文件，关键参数设置：

python train.py

--epochs 30

--batch\_size 32

--lr 0.0001

--data\_root /data/data

1. 模型测试：运行test.py生成显著图，结果保存至/results目录：

python test.py --model\_path scribble\_30.pth --save\_root results

五、数据集

5.1 数据集介绍

S-DUTS 数据集：基于基准显著目标检测数据集 DUTS，通过人工标注生成三种涂鸦标注版本（细涂鸦、宽涂鸦），包含 10,553 张训练图像和 5,019 张测试图像，适用于弱监督 SOD 模型训练与评估。

5.2 数据预处理

1. 格式转换：使用 Matlab 脚本完成数据转换：
   * generate\_gt\_mask\_from\_scribble.m：将涂鸦标注转换为二进制掩码（gt）和辅助掩码（mask）；
   * convert\_rgb2gray.m：将 RGB 图像转为灰度图，降低计算复杂度。
2. 数据加载：通过data.py中的SalObjDataset类和test\_dataset类实现数据加载，支持数据增强（随机裁剪、翻转）和归一化处理，确保数据符合模型输入要求。

六、消融实验对比

6.1 实验目的

通过逐步移除或替换算法中的注意力模块、ASPP模块和损失函数（平滑损失），验证各组件对模型性能的贡献，明确算法的核心优势与改进方向。

6.2 实验设置

**1.对比模型设置**

|  |  |  |
| --- | --- | --- |
| 模型版本 | 移除的组件 | 核心特点 |
| 原始模型（Scribble\_Saliency） | - | 完整结构，包含 ASPP、注意力模块、平滑损失 |
| 模型 A | ASPP 模块 | 仅保留基础编解码结构，移除多尺度特征融合 |
| 模型 B | 注意力模块 | 移除边缘细节增强模块 |
| 模型 C | 平滑损失（smoothness\_loss） | 仅使用二元交叉熵损失 |

**2.实验配置**

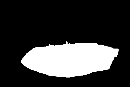
1. 数据集：使用S-DUTS测试集（包含 5019 张图像），覆盖复杂背景、小目标等场景。
2. 评估指标：F-measure（平衡精确率与召回率）、MAE（平均绝对误差）、E-measure（增强对齐度量）。​
3. 训练参数：batch\_size=32，学习率=1e-4，训练30轮，使用相同的初始化权重和数据增强策略。

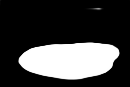
6.3 实验结果

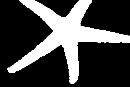
1. **量化指标对比**：

|  |  |  |  |
| --- | --- | --- | --- |
| 模型版本 | F-measure | MAE | E-measure |
| 原始模型 | **0.8878** | **0.0617** | 1.2896 |
| 模型 A（无ASPP） | 0.3042 | 0.5555 | 0.2502 |
| 模型 B（无注意力） | 0.8061 | 0.0996 | 0.9531 |
| 模型 C（无平滑损失） | 0.8515 | 0.0892 | 0.9787 |

1. 可视化结果对比

山上的风景

AI 生成的内容可能不正确。

图片包含 室内, 看着, 黑暗, 前

AI 生成的内容可能不正确。图片包含 游戏机, 大, 沙滩, 海

AI 生成的内容可能不正确。图片包含 游戏机, 灯光

AI 生成的内容可能不正确。图片包含 游戏机, 灯光

AI 生成的内容可能不正确。

图片包含 游戏机, 交通, 齿轮, 轮子

AI 生成的内容可能不正确。图片包含 看着, 火, 前, 黑暗

AI 生成的内容可能不正确。小鸟

AI 生成的内容可能不正确。图片包含 游戏机

AI 生成的内容可能不正确。卡通人物

AI 生成的内容可能不正确。

卡通人物

AI 生成的内容可能不正确。草地上有只黑白色的照片

AI 生成的内容可能不正确。卡通人物

AI 生成的内容可能不正确。图片包含 游戏机

AI 生成的内容可能不正确。

* + 1. Img (2)GT (3)原始模型 (4)模型A (5)模型B (6)模型C

6.4 结果分析

|  |  |  |  |
| --- | --- | --- | --- |
| 模块缺失 | 核心影响 | 指标变化特征 | 可视化典型问题 |
| ASPP（模型 A） | 多尺度特征融合能力崩塌 | 所有指标暴跌（F 降 58%，MAE 升 7.7 倍） | 目标丢失、破碎、背景误判 |
| 注意力（模型 B） | 特征判别力下降 | 指标缓降（F 降 9%，MAE 升 58%） | 边缘模糊、细节丢失 |
| 平滑损失（模型 C） | 空间连续性约束消失 | 指标微降（F 降 4%，MAE 升 45%） | 噪点、区域破碎 |

**模块价值排序与优化启示**

模块优先级：**ASPP>注意力>平滑损失**

**优化方向：**

1. 模型 A 的教训：若需轻量化，不可移除 ASPP，可尝试 “轻量化 ASPP”（如减少空洞卷积分支数，或替换为动态多尺度模块）。
2. 模型 B 的潜力：改进注意力机制（如引入 动态注意力，根据目标尺度自适应调整权重），进一步提升特征判别力。
3. 模型 C 的优化：设计更高效的 结构约束损失（如结合边缘检测的损失函数），替代简单的平滑损失，在减少噪点的同时增强边缘精度。

七、总结

7.1. 核心成果

1. 弱监督有效性验证：通过涂鸦标注替代像素级标注，在 S-DUTS 数据集上实现F-measure 0.8878，降低标注成本的同时优于部分全监督模型，证明弱监督方法在显著检测中的可行性。
2. 关键模块价值：消融实验明确 ASPP 模块（多尺度特征融合）是复杂场景检测的基础（缺失导致 F-measure 暴跌 58%），Edge\_Module（边缘增强）和平滑损失（空间约束）分别提升轮廓精度与视觉质量。
3. 工程落地实践：完成从本地到 GPU 云服务器的环境迁移，实现数据预处理、模型训练与测试的全流程部署，为算法规模化应用奠定基础。

7.2. 技术瓶颈与优化方向

1. 复杂场景适应性：当前模型对小目标和高相似背景场景表现较弱，未来可引入自监督预训练或动态多尺度机制强化特征表达。
2. 轻量化与实时性：基于 VGG16 的骨干网络计算量较大，计划替换为 MobileNet 等轻量化架构，结合模型蒸馏优化边缘模块与 ASPP 结构，平衡精度与效率。
3. 标注效率提升：探索交互式标注引导和关键位置自动识别算法，充分利用涂鸦标注的稀疏性，减少人工标注成本。

7.3. 实践价值与未来展望

项目验证了弱监督检测的技术价值，积累了从算法设计到工程实现的全流程经验。未来将聚焦高效弱监督框架研究，推动显著检测技术在智能终端、医疗影像等领域的落地应用，兼顾精度、效率与标注成本的平衡。