

学习计划（大创）

Created @2024年9月19日 20:18

大创计划9.19

- 跟语义信息结合在一起，立面要识别要素，open vocabulary语义分割
- 书生大模型internVL，源代码里面实际上有图生图的部分，比较靠谱
- 输入打包的压缩包，解压后有obj,mtl,纹理，服务器可以根据图片做一些系统化的处理，重点是二维效果，前端用threejs即可，先搞一栋建筑物
- 用训练好的模型即可，awesome...用clip模型比较老，搞一个新一点的模型结合
- 感觉需要先做出一个页面，看看智谱怎么用（？）
- 链式结构，langchain？



要进行基于AIGC的城市级建筑物三维模型纹理生成与辅助修复，结合“open vocabulary 语义分割”、“图生图”、“LangChain”等概念，可参考思路和步骤如下：

1. 整体规划：基于AIGC的城市级建筑物三维模型纹理生成与辅助修复框架

需要处理的任务主要包括：

- 语义分割**：从建筑物的图像或三维数据中识别出关键要素，比如窗户、门、墙体等。
- 三维纹理生成**：从图像或模型中生成相应的纹理，应用到三维模型上。
- 辅助修复**：在模型生成时自动识别损坏或缺失的部分，并进行修复。

这些任务可以通过一个系统化的流程来完成，包含以下主要步骤：

2. 数据处理与输入

输入文件：

- 压缩包包含 `.obj` 文件（用于三维模型）、`.mtl` 文件（材质文件）、以及纹理图片。你需要将这些文件解压，解析出三维模型的结构，并将纹理应用于模型。

处理思路：

- **解压与文件解析：**可以使用 Python 脚本来处理上传的压缩文件，提取 `.obj`、`.mtl`、和纹理图像文件。
- **预处理：**对建筑物模型和纹理图片进行处理，确保图片分辨率、光照等一致性。可以使用 Open3D 和 Pillow 等库来读取和调整三维模型和图像。

相关工具：

- Python 的 `zipfile` 模块可处理压缩包的解压。
- `Open3D` 库处理三维模型的读取、显示与编辑。
- `Pillow` 库用于处理图像数据。

3. 模型选择与集成：新型的AIGC与Open Vocabulary语义分割模型

你的需求涉及语义分割（open vocabulary）和图生图（图像生成），CLIP 虽然比较老，但仍然是重要的基石模型。你想用更新的模型，因此可以考虑以下几种现代模型，尤其是针对视觉语义理解的新型多模态模型：

模型建议：

- **InternVL：**书生大模型的 InternVL（图像-文本多模态模型）结合了视觉语义信息，适合处理图像到三维数据的映射与生成任务。它可以结合视觉与文本描述，进行开放词汇的语义分割和理解。利用 InternVL，你可以识别建筑物中的不同要素，比如门、窗、外墙等。
- **CLIP Variants（如 CLIP-GPT 模型）：**比经典 CLIP 更强大的变体模型可以处理开放词汇语义分割的任务，适合用于结合建筑图像中的语义要素来指导纹理生成与修复。
- **Segment Anything Model (SAM)：**用于语义分割的最新模型，能够实现高度自动化的分割处理。
- **Stable Diffusion XL 或者 DALL-E 3：**用于图生图任务，可以在获取建筑物的初步语义理解后，生成或者修复三维模型的纹理图像。

Streamlit

 <https://internvl.opengvlab.com/>

internVL似乎包含图生图

<https://github.com/Qinying-Liu/Awesome-Open-Vocabulary-Semantic-Segmentation>

这个 GitHub 项目 "**Awesome-Open-Vocabulary-Semantic-Segmentation**" 是一个开源资源库，专门收集和整理关于 **开放词汇语义分割（Open-Vocabulary Semantic Segmentation）** 领域的最新研究、算法、代码库和文献。其主要功能是为研究者、开发者提供一站式的参考，帮助他们理解和应用这些前沿技术。

1. 核心作用

1.1 开放词汇语义分割（Open-Vocabulary Semantic Segmentation）

开放词汇语义分割的核心任务是：

- 让模型可以分割并识别图像中的像素级语义标签，而不局限于训练集中预定义类别。
- 模型能够通过自然语言描述（通常使用视觉-语言模型）来处理以前未见过的类别。

该项目整理了从**全监督**、**弱监督**到**无训练**方法的多个研究成果，并附带了每个方法的论文和代码链接。这意味着开发者可以基于不同场景需求，选择合适的分割方法或模型，快速上手进行研究或项目开发。

1.2 研究前沿汇总

该项目定期更新，涵盖了最新的会议如 CVPR、ECCV、ICML 等会议上的论文。因此，它可以起到：

- **快速获取新技术**：研究人员可以通过该项目了解最前沿的开放词汇语义分割技术。
- **代码实现参考**：它包含了大量附带代码的模型（如 CLIP、SegCLIP、FreeSeg 等），研究者可以直接下载并使用这些代码，进行实验和模型微调。

1.3 跨任务扩展

项目不仅限于开放词汇语义分割，还扩展到了与语义分割相关的其他任务，比如：

- **零样本语义分割 (Zero-shot Semantic Segmentation)**：用于处理在训练过程中从未见过的类别。
- **引用图像分割 (Referring Image Segmentation)**：通过自然语言指定图像中的某一部分进行分割。
- **开放词汇目标检测 (Open-Vocabulary Object Detection)**：识别图像中的物体，同时处理那些没有标注的类。
- **very important: 语义信息！** 要首先识别窗户、屋顶、地面这种部分，这里的open vocabulary理论上可以直接告诉模型需要识别的东西的单词

1.4 模型的无训练方法

项目还列出了许多无训练 (Training-Free) 的开放词汇语义分割方法。它们是通过直接修改预训练模型（如 CLIP 或扩散模型）来实现分割，而无需额外的训练。对于开发者来说，这提供了一种快速原型设计的方式，因为不需要耗费大量的时间和资源进行模型训练。

2. 潜在应用场景

这个项目的资源可以广泛应用于以下场景：

2.1 增强语义分割系统

如果你正在开发一个需要处理动态场景或新类别的语义分割系统（例如建筑物、城市场景或自动驾驶），这些开放词汇语义分割技术可以帮助你：

- **识别新的建筑元素**：即使你的训练数据集没有特定的建筑构件（如雕像、古建筑风格的门窗），这些模型可以通过开放词汇识别出这些元素。
- **泛化能力更强**：传统语义分割通常限制于训练数据集中的类别，而开放词汇模型可以处理更多类别，特别是使用 CLIP 等多模态模型结合文本描述进行分割时。

2.2 结合三维模型与图像的应用

结合你之前提到的三维建筑物纹理生成与修复任务，这个项目中的技术可以帮助：

- **图像中的纹理要素分割**：通过语义分割，你可以自动从建筑物图像中提取特定纹理要素（如墙体、窗户、门等），并将它们应用于三维模型的修复中。
- **开放式类别扩展**：如果建筑物有未知或未标记的部分，开放词汇模型可以通过自然语言描述进行分割，并生成相应的纹理或补全三维模型中的缺失部分。

2.3 弱监督学习场景

在资源有限的情况下（如没有大量标注数据），你可以利用项目中的弱监督或无监督方法进行语义分割和模型训练，显著减少标注成本。例如：

- **只通过图片和文本对进行训练：**弱监督方法可以从图像和描述性文本中学习分割，而不需要逐像素的精确标注。

2.4 跨领域应用

这些模型不仅限于城市级三维建筑物处理，它们可以应用在更广泛的场景中，如：

- **医疗图像分割：**你可以使用这些开放词汇模型来分割和识别医学图像中的未知病变或器官。
- **卫星图像分割：**可以应用于识别卫星图像中的地形特征，比如未标注的城市扩展区域或森林火灾区域。

3. 可视化和前端应用

你提到的前端应用场景（如 Three.js）也可以结合这个项目中的模型来实现：

- **三维模型与语义分割的结合：**通过 open-vocabulary segmentation，前端可以通过自然语言或图像描述来动态更新三维建筑模型，进行语义高亮或部分修复。
- **交互式工具开发：**用户可以通过自然语言或文本框输入描述（如“标记所有窗户”），系统通过这些语义分割模型自动在三维模型上高亮出相关区域。

4. 总结

这个项目的主要作用是：

1. **提供开放词汇语义分割技术的研究前沿和实现**，可以帮助开发者快速掌握、应用最新的模型。
2. **丰富的代码资源**，包括从全监督到无训练的模型，实现简单、易用的开放式分割。
3. **为弱监督和无监督学习提供解决方案**，特别是在标注数据有限的情况下。
4. **广泛应用于语义分割、目标检测、图像生成等任务**，可以为你的三维建筑物处理与纹理生成项目提供模型和技术支持。

结合你提到的需求，该项目可以为你的工作提供开箱即用的模型，减少开发时间，同时提升系统的智能化水平。

Streamlit

 <https://internvl.opengvlab.com/>

4. 系统架构与前端展示：three.js + langchain

三维模型显示与前端展示：

- **Three.js**：你可以利用 Three.js 作为前端的三维模型显示工具，它支持浏览器中的 3D 图形渲染，适合用于建筑物三维模型的展示和纹理应用。

LangChain 的链式结构集成：

LangChain 是用于构建基于语言模型（LLM）应用的工具链，它适合用于：

- **任务分配与模型交互**：LangChain 可以帮助你构建一个多步骤的任务链，用于调用不同的 AI 模型处理不同阶段的任务。例如，先进行语义分割、然后是纹理生成，再到三维模型修复。LangChain 可以整合这些模块化的任务，并通过自然语言或特定指令进行链式调用。
- **输入与输出的转换**：通过 LangChain，将用户输入的自然语言描述（比如某建筑物需要修复或优化的部分）转换成可以供不同模型调用的具体指令。

图生图部分：

- **InternVL 或 Stable Diffusion** 模型可以用于生成修复后的建筑纹理图片。通过与三维模型的语义分割相结合，这些模型可以生成符合实际需求的建筑物外观纹理。

5. workflow 设计：从输入到模型生成的流程

1. 数据上传与解析：

用户上传包含三维模型与纹理的压缩包，通过 Python 解压并解析出 `.obj` 和 `.mtl` 文件。

2. 语义分割与识别：

使用 InternVL 或 Segment Anything Model（SAM）等进行开放词汇语义分割，识别出建筑物的重要要素（门、窗、墙等）。

3. 三维纹理生成：

根据分割出的建筑要素，调用图生图模型（如 Stable Diffusion、DALL-E 3）生成新的纹理，或者根据损坏部分进行修复。

4. 辅助修复与细节完善：

在生成的纹理上，结合多模态的图像与文本模型对缺失或损坏部分进行自动化修复与补充。

5. 前端展示：

通过 Three.js 将建筑物三维模型及生成的纹理效果展示在前端，允许用户进行交互式查看。

6. 模型微调与扩展

- 如果现有模型的效果不完全符合你的需求，可以考虑对这些模型进行微调。尤其是对于建筑物语义分割和纹理生成部分，你可以使用自己收集的建筑物图像和三维模型数据进行微调，提升特定任务上的表现。
- **云端部署：**将模型和处理流程部署到云端服务器，方便用户上传文件并实时获取处理结果。可以使用 GPU 实例（如 AWS、Google Cloud）加速模型推理。

7. 智谱大模型的应用

可以考虑将智谱的多模态模型与 InternVL 或类似的大模型相结合，进一步优化语义分割和生成的效果。智谱的多模态能力可以用于提升建筑物模型理解与生成的精度，特别是在处理大规模城市级项目时，它的性能会非常重要。

总结：

根据你的需求和现有的技术趋势，可以遵循以下步骤：

1. **数据解析与三维模型加载：**解压并读取三维模型与纹理图像。
2. **语义分割与要素识别：**使用 InternVL、SAM 等模型进行开放词汇语义分割。
3. **三维纹理生成与修复：**利用图生图模型生成或修复三维模型的纹理。
4. **前端展示：**使用 Three.js 、electron进行交互式的三维模型展示。
5. **链式任务执行：**用 LangChain 等工具整合模型调用与任务分配。

生成纹理算法

☒ ~~学习一下trimesh,pyvista两个库，了解obj,mtl文件格式→wiki，看看如何把纹理映射到立面上~~

☐ 建筑物立面处理，style transfer

☐ 了解图像补全(inpainting)和图像生成的研究进展和现状，参考来源：综述性文章或者知乎

☐ 一些树会挡住建筑物，去掉树(?)然后填充被挡住的模糊的地方 inpainting (局部重绘)

☐ 结合语义分割，图像重绘实现自动化识别树木区域并对树木区域进行重绘

☐ 这些技术对应的算法代码需要尝试能否用在自己的数据集(?)然后看是否能改写一下

- 组合别人训练好的模型，实现工程
- 通过综述文章找到经典或比较火的算法，跑通代码

Stable diffusion Model

python脚本：以stable diffusion为基础，先多尝试一些提示词来优化已有的图片（建筑物立面）

图生图⇒根据输入的建筑物图片（也可以是一次性多个建筑物），首先由模型自己生成一段场景描述（利用开源算法），利用模型优化重组，作为新的提示词生成新的图
语义分割 识别建筑物侧面遮挡，去掉涉及到树的像素.....

PART1（小学期结束前实现）

Trimesh,Pyvista学习：

☒ ~~了解obj,mtl文件格式，可以看一下wiki（文本格式，用trimesh画出一个立方体，导出，并对这六个面每个面设置一个不同的颜色，一行一行看一下格式）~~

☒ 手动创建obj文件，设置颜色并尝试将纹理映射到面上

