基于 SAM 的图像篡改

杨轶超

2024 年 1 月 13 日

摘要

本次课程实验，学习、了解并使用最新的图像分割领域成果 SAM，将 SAM 用于图像篡改，并通过数据增强实现多种篡改，为后续继续完善篡改脚本，生成未来用于篡改检测的一种预训练数据。

关键词：图像篡改；数据增强；图像分割

1 引言

随着深度学习尤其是对抗生成网络的发展，图像编辑应用正逐渐走进大众的日常生活。然而，这些被修改的图片的视觉质量愈发提高，甚至可以达到以假乱真的效果。这就为互联网信息的可信性带来了严峻的挑战。为了后续研发对于图像篡改检测的方法，就需要脚本批量生成一些篡改图像，为后续的预训练提供数据。

在 2023 年由 Facebook 发布了 Segment Anything（SA）项目 [2]，很快成为 2023 年计算机视觉领域、分割任务领域重大的成果。本文想尝试将该项目提出的分割一切模型（Segment Anything Model, SAM）应用于图像篡改，生成一些篡改图像为之后的篡改检测研究做准备。

2 相关工作

2.1 图像分割

图像分割任务旨在鉴别区分出一张图片的不同部分，比如人物、汽车等等。从技术角度讲，图像分割任务需要根据不同的语义信息区分并聚集起对应相同语义的像素点。大体上，图像分割可以分为三个子任务：实例分割 (Instance Segmentation) 、语义分割 (Semantic Segmentation) 、全景分割 (Panoptic Segmentation)。这三个子任务都有着大量的算法与模型。这之中的语义分割，通俗解释是让计算机根据图像的语义来进行分割，是一种像素级的分割。

在深度学习应用到计算机视觉领域之前，研究人员一般使用纹理基元森林 (Texton Forest) 或是随机森林 (Random Forest) 方法来构建用于语义分割的分类器。

随着深度学习越来越多的应用于计算机视觉领域，出现了很多语义分割模型。较为经典的有：2014 年，加州大学伯克利分校的 Long 等人提出的完全卷积网络 (Fully Convolutional Networks)，推广了原有的 CNN 结构，在不带有全连接层的情况下能进行密集预测。 [3] 在

这之后，研究人员为了解决影响 CNN 对于分割问题使用的池化层问题，提出两种方法解决：编码器-解码器 (encoder-decoder) 结构 [1] 和采用空洞卷积结构 [5]，去除池化层结构。

2.2 图像篡改

图像篡改就是通过各种方式，包括但不限于 PS 修图，AI 扩图等，修改图像的内容、颜色、明亮度等等。

一般情况下认为主要的图像篡改方法大致可以分为三种：①拼接（Splicing）将一张图像上的内容粘贴到另一张图像上；②复制-粘贴（Copy-move）将一张图像上的内容复制到同图的其他位置;③移除（Removal）将图像上的某个区域去掉并根据周围像素进行修复。 [4]

其中不难发现，拼接和复制-粘贴的最主要区别就是在同一张图像上进行操作还是不同图像，而移除相对特殊，需要对移除区域的像素进行具有一定相似性的修复。

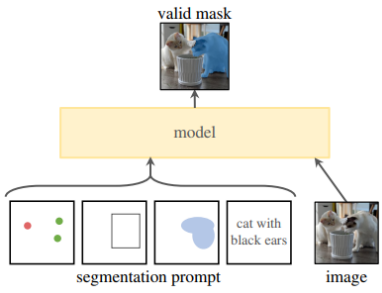
3 本文方法

3.1 本文方法概述

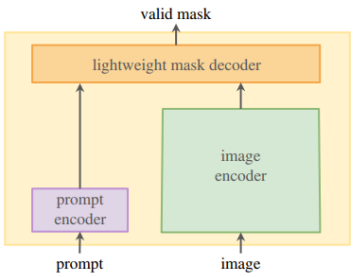
预训练的大语言模型在 NLP 领域引发了一场革命，因为它们具有强大的零样本（zero-shot）和少量样本（few-shot）泛化能力，能够泛化到训练期间未见过的任务和数据分布。这些“基础模型”通常通过提示工程（prompt engineering）来实现，其中手工制作的文本用于提示语言模型为任务生成有效的文本响应。当规模扩大并使用来自网络的丰富文本语料库进行训练时，这些模型的零样本和少样本性能令人印象深刻，通常与微调模型相当。经验趋势表明，随着模型规模、数据集大小和总训练计算量的增加，这种行为会有所改善。

SA 的目标是为图像分割领域构建一个类似的基础模型。这意味着 SA 正在寻求开发一个可提示的模型，并在广泛的数据集上使用一项任务进行预训练，以实现强大的泛化能力。通过这个模型，SA 的目标是利用提示工程来解决一系列下游分割问题，而无需重新训练模型。

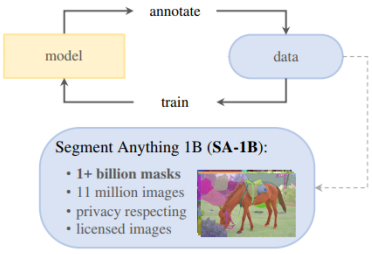
这项计划的成功与否和三个组成部分密切相关：任务、模型和数据。就是要解决三个问题：什么样的任务可以实现零样本泛化? 对应的模型架构是什么? 什么数据可以支撑这项任务和模型? 为此 SA 提出了三个问题的回答，分别是可提示的分割任务，如图 1a所示，分割一切模型，如图 1b所示，数据引擎和数据集，如图 1c所示。 [2]



(a) **Task**: promptable segmentation



(b) **Model**: Segment Anything Model (**SAM**)



(c) **Data**: data engine (top) & dataset (bottom)

图 1. Segment Anything 解决的三大问题

3.2 可提示分割任务

作者从 NLP 自然语言处理中获得灵感，其中下一个 token 预测任务用于基础模型预训练，并通过提示工程来解决各种下游任务。为了构建分割的基础模型，作者的目标是定义具有类似功能的任务。

任务 Task：作者首先将 NLP 领域中的提示工程思想应用到图像分割领域中。一个提示可以是前景/背景点、粗略的框或掩码，或者任何指示在图像中区分什么的信息。可提示分割任务的目标是，给定任何提示，返回一个有效的分割掩码，而掩码的“有效”性只是要求它至少包含提示中所指示的一些对象。这一要求类似于语言模型对模糊提示的期望连贯响应。作者选择这个任务是为了实现一种自然的分割预训练算法，并通过提示将可提示的分割任务迁移到各种下游分割任务中。

预训练 Pre-training：可携带的分割模型提出了一种通用的预训练方法，即为每个训练样本生成一系列提示（例如点、框、掩码），并将掩码预测与真值的 GroundTruth 比较。作者借鉴了交互式分割的方法，尽管目标是在足够用户输入后最终生成有效的掩码，但始终预测有效的掩模以确保预训练的模型适用于涉及模糊性的场景，包括自动注释的需求。作者指出，执行这个任务具有挑战性，需要专门的建模以及选择适当的训练损失。 [2]

3.3 分割一切模型 SAM

分割一切模型主要有三部分构成，分别是图像编码器，灵活的提示编码器和快速的掩码解码器，如图 2所示。

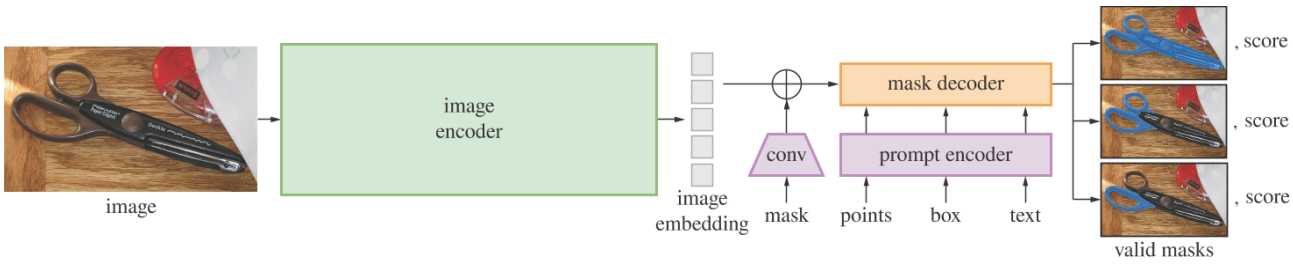


图 2. 分割一切模型 SAM 示意图

图像编码器 Image encoder：受到可扩展性和强大的预训练方法的启发，我们使用了一个 MAE 预训练的 Vision Transformer（ViT），最小化地适应处理高分辨率输入。图像编码器每个图像运行一次，并且可以在提示模型之前应用。

提示编码器 Prompt encoder：考虑两组提示：稀疏（点，框，文本）和密集（掩码）。对于点和框，我们使用位置编码来表示它们，并将这些编码与每个提示类型的学习嵌入式相加。我们使用 CLIP 的文本编码器来代表自由文本，而密集提示（即遮罩）则使用卷积嵌入式，并与图像嵌入式元素相加。

掩码解码器 Mask decoder：掩码解码器通过有效地将图像嵌入，提示嵌入和输出 token 映射到掩码来实现。这种设计采用了一个修改的 Transformer 解码器块，后跟一个动态掩码预测头。修改的解码器块使用提示自注意力和交叉注意力两个方向（提示到图像嵌入和反之亦然）来更新所有嵌入。在运行两个块之后，对图像嵌入进行上采样，并且 MLP 将输出令牌映射到动态线性分类器，然后在每个图像位置计算掩码前景概率。 [2]

4 复现细节

4.1 与已有开源代码对比

对于 SAM 的部分，使用 Facebook 提供的开源代码并未有任何修改。对于图像篡改部分，通过 SAM 获取的 mask 值使用 pytorch 篡改图像，并使用 torchvision.transforms 进行数据增强后篡改图像。

4.2 实验环境搭建

实验环境，如表 1所示。虽然 SAM 要求 Python 版本大于等于 3.8，但由于与前辈师兄对接方便采用 3.7 版本，总体而言不影响使用。

CPU	Intel(R) Xeon(R) CPU E5-2695 v4 @ 2.10GHz
GPU	Nvidia Tesla P100-PCIE-16GB
操作系统	CentOS Linux release 7.2
编程语言	Python 3.7
主要第三方库	PyTorch 1.12
集成开发环境	PyCharm 远程开发

表 1. 实验软硬件环境

在上述平台，直接执行 Facebook 提供的 setup 程序，测试试用 demo，正常运行，环境搭建完毕。

4.3 创新点

SAM 作为最新的图像分割领域成果，可以将其应用于多个方面。在图像篡改方面，也需要对于图像进行分割后进行篡改操作。本次实验是一个很好的尝试将最新的图像分割领域成果应用于图像篡改。

5 实验结果分析

本次实验首先随机选择一个点位，通过 SAM 获取以该点为 prompt 的三个 mask 值，随机选取其中一个，如图 3a所示，这是选取了该点获取的第三个 mask 值。

通过使用 PyTroch 提供的 masked 方法就可以将图 3a获取的纸袋拼接到小狗图中，如图 3b所示。同时可以对获取到的 mask 进行操作，实现篡改目标位移，如图 3c所示。

然后我们对原图进行了数据增强，以使得篡改目标可以翻转、旋转、放缩。图 3d和图 3e分别由原图及其获取点位进行横、纵向翻转得到。图 3f和图 3g分别由原图及其获取点位进行顺、逆时针旋转得到。图 3h和图 3i分别由原图及其获取点位进行放大、缩小得到。

(a) 获取篡改目标　(b) 对应位置篡改　(c) 篡改位置评议

(d) 篡改目标横向翻转　(e) 篡改目标纵向翻转　(f) 篡改目标逆时针旋转 45 度

(g) 篡改目标顺时针旋转 45 度　(h) 篡改目标放大　(i) 篡改目标缩小

图 3. 图像篡改尝试示例

6 总结与展望

在本次实验中，我学习了计算机视觉领域的图像分割任务，并了解了图像篡改相关的基本情况。我尝试了一些篡改方式，包括拼接、复制-贴入，并通过对源图像进行缩放、旋转和翻转等操作来实现多种篡改。

然而，我也意识到处理篡改图像时存在的不足之处。首先，我还没有尝试过使用移除的篡改方式，因此我计划在未来继续尝试这些方法。另外，我目前的篡改尝试是手动进行的，因此我计划将程序修改为自动化脚本，以提高篡改尝试的效率和准确性。最后，我还没有对篡改边缘进行处理，因此我计划尝试在篡改边缘添加噪声等方式，以使篡改效果更好。

我希望通过以上这些计划能够完善本次实验的篡改内容，最终形成自动化的脚本，为后续的篡改检测提供一组经过预训练的数据，从而更好地保护我们的信息和安全。

参考文献

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.

[2] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, October 2023.

[3] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):640–651, 2017.

[4] Junke Wang, Zuxuan Wu, Jingjing Chen, Xintong Han, Abhinav Shrivastava, Ser-Nam Lim, and Yu-Gang Jiang. Objectformer for image manipulation detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2364–2373, June 2022.

[5] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.