

Open-Vocabulary Semantic Segmentation with Mask-adapted CLIP

摘要

开放词汇语义分割旨在根据文字描述（模型在训练过程中可能没有看到过这些文字描述）将图像划分为语义区域。最近的两阶段方法首先生成类不可知的掩码提议，然后利用预训练的视觉-语言模型，如 CLIP，对掩码区域进行分类。我们将该范式的性能瓶颈确定为预训练的 CLIP 模型，因为它在掩膜图像上表现不佳。为了解决这个问题，我们提出在一组掩膜图像区域及其对应的文字描述上对 CLIP 进行微调。我们通过挖掘现有的图像-描述数据集（例如，COCO Captions）收集训练数据，使用 CLIP 将被遮掩图像区域与图像描述中的名词进行匹配。与使用固定类别（例如，COCO-Stuff）的更精确和手动标注的分割标签相比，发现我们收集的噪声多但多样化的数据集可以更好地保留 CLIP 的泛化能力。在对整个模型进行微调的同时，我们还利用掩膜图像中的“空白”区域，进行了掩码提示调整（mask prompt tuning）。实验表明，在不修改 CLIP 的任何权重的情况下，掩码提示调整带来了显著的改进，并且它可以进一步改进完全微调的模型。特别地，当在 COCO 上训练并在 ADE20K150 上评估时，我们的最佳模型达到了 29.6 % mIoU，比之前的模型提高了 + 8.5 %。在没有数据集特定适应性的情况下，开放词汇的通用模型首次匹配了 2017 年提出的有监督专家模型的性能。

关键词：开放词汇；语义分割；CLIP

1 引言

语义分割旨在将像素分组为具有相应语义类别的有意义区域。尽管其取得了令人瞩目的进展，但现代语义分割的模型主要使用预先定义的类别进行训练，无法泛化到未见的类别。相反，人类以开放词汇的方式理解场景，通常有数千个类别。为了接近人类层面的感知，近些年来新兴的开放词汇语义分割可实现通过文本描述对图像中的任意类别进行语义分割，因此选择了此篇来自于 CVPR2023 且效果优异的文章 [18] 来进行复现。

2 相关工作

2.1 预训练的视觉-语言模型

预训练的视觉-语言模型 [11, 17, 22, 23] 将视觉与文本描述联系起来。预训练的 CLIP [22] 具有很强的开放词汇分类能力，即可根据语言描述的任意类别对图像进行分类。预训练的 CLIP

赋予了许多计算机视觉任务能够使用语言的能力，如图像处理 [2]、图像生成 [5]、目标检测 [10, 28] 和图像分割 [6, 7, 9, 13, 15, 20, 26, 27] 等。本篇文章的工作类似于 RegionCLIP [28]，通过对候选区域进行微调，将 CLIP 应用于目标检测。文章使用的方法与 RegionCLIP 的不同之处在于：(1) RegionCLIP 用于处理完整裁剪下来的区域，而这篇文章则使用 CLIP 来处理掩膜图像；(2) 利用掩膜图像中的空白区域，提出掩码提示调整策略，该调整可在不改变其权重的情况下适应 CLIP。这使得在多任务场景中可以与其他任务共享 CLIP 的权重，而 RegionCLIP 不支持此功能。

2.2 开放词汇分割

开放词汇分割旨在理解具有文本描述的任意类别的图像。开创性工作 ZS3Net [3] 使用生成模型通过未见过的类的词嵌入来合成像素级特征。SPNet [24] 利用词嵌入，像 word2vec [21]，将语义与视觉特征对齐。GroupViT [25] 直接从文本监督中对分割掩码进行分组。最近，研究人员建议利用预先训练的 CLIP [22] 进行开放词汇语义分割。LSeg [15] 将像素嵌入与相应语义类的文本嵌入对齐，该语义类由 CLIP 的文本编码器生成。与像素级 LSeg 不同，OpenSeg [9] 提出通过区域词基础将分段级视觉特征与文本嵌入对齐。本文的方法属于两阶段方法的范畴，例如 ZSSeg [27] 和 ZegFormer [6]：它们首先生成与类别无关的掩码建议，然后利用预先训练的 CLIP 来执行开放词汇分类。与直接使用原始 CLIP 进行掩膜图像分类的 ZSSeg 和 ZegFormer 不同，这篇文章通过调整 CLIP 来提高性能。

2.3 提示调整 (Prompt tuning)

提示调整是一种使大规模预训练模型适应新任务的策略。这个想法起源于自然语言处理 [14, 16, 19]，最近的工作将提示调整扩展到计算机视觉。CoOp [29] 在类别词前添加可学习向量，以使 CLIP 适应许多识别任务。文本的提示调整也广泛应用于开放词汇对象检测 [8] 和语义分割 [27]。我们的掩码提示调整与视觉域中的提示调整更相关 [1, 12]，其中可学习向量应用于图像域。与在实际图像标记之前插入附加标记的视觉提示调整 [12] 不同，这篇文章用可学习的提示替换屏蔽标记。此外，掩码提示调整比完全微调的模型带来了额外的改进。先前的工作尚未报告此类额外的改进。

3 本文方法

3.1 本文方法概述

文章提出的两阶段开放词汇语义分割模型如图 1 所示。它由生成掩码提案的分割模型和开放词汇分类模型组成。

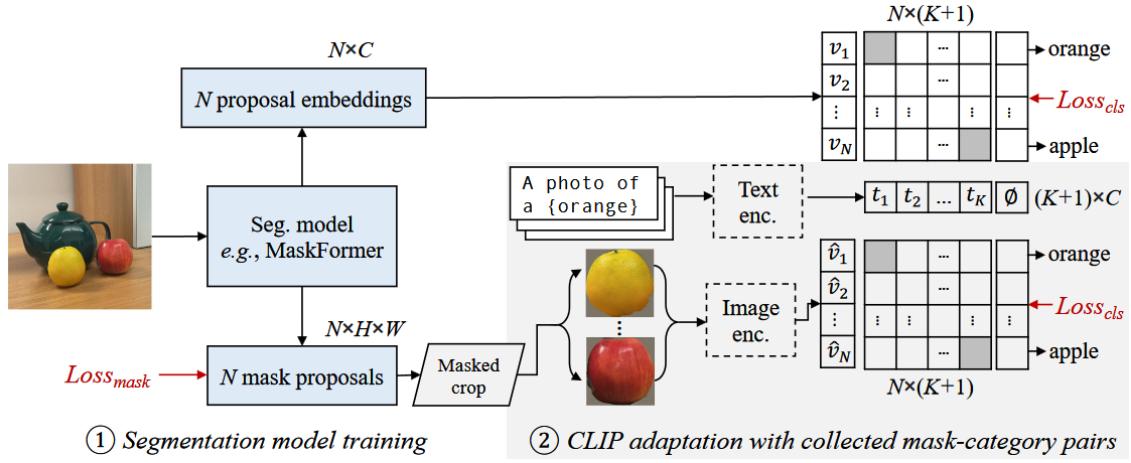


图 1. 两阶段方法模型

两阶段方法由一个分割模型 MaskFormer 和一个 CLIP 模型组成。首先，经过修改后的 MaskFormer 使用 CLIP 的文本嵌入进行训练，以执行开放词汇分割。然后，使用预先训练的分割模型来生成与类别无关的提案，并将提案与从相应标题中提取的名词进行对齐。收集不同的掩码-类别对后，使用建议的掩码提示调整对 CLIP 进行微调。

3.2 从图片描述中收集不同的掩码-类别对

为了使 CLIP 更好地处理掩膜图像，文章建议在由掩膜图像和文本对组成的数据集上微调 CLIP。一种直接的解决方案是利用手动注释的分割标签，例如来自 COCO-Stuff 的标签，此类标签是准确的，但是一组封闭的类别。然而，作者观察到这种简单的方法限制了 CLIP 的泛化能力，性能有所下降。作者认为这是由于文本词汇量有限，微调后的 CLIP 会过度拟合这 171 个类别，从而失去泛化到未见过的类别的能力。

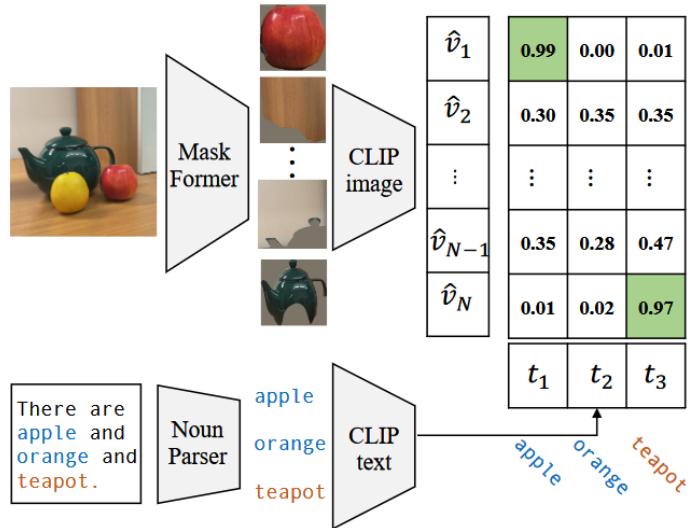


图 2. 对于给定的图像-标题对，COCO 数据集中只有“苹果”和“橙色”是类别。通过从标题中提取名词，我们还可以得到一个新的“茶壶”类别。

作者的做法是设计一种自标记策略 [9, 28] 来提取掩码-类别对。如图 2 所示，给定图像，我们首先使用预先训练的 MaskFormer 来提取掩码提案。同时，我们使用现成的语言解析器 [2]

从相应的图像标题中提取所有名词，并将它们视为潜在的类。然后，使用 CLIP 将最匹配的掩码提案与每个类别配对。从 COCO-Captions [4] 中，作者每个图像使用 5 个标题，共收集了 1.3M 个掩码-类别对，其中包含 27K 个名词类别；或者每个图像使用 1 个标题，共收集了 440K 个掩码类别对，其中包含 12K 个名词类别。实验表明，这种嘈杂但多样化的掩膜类别数据集比手动分割标签具有明显更好的性能。

3.3 掩码提示调整

掩膜图像和自然图像之间最显着的区别是掩膜图像中的背景像素设置为零，导致出现许多“空白区域”。当将掩膜图像输入 CLIP 时，图像将被分为互不重叠的块，然后分别对每块进行标记化。这些空白区域将变成零标记。这些标记不仅不包含有用的信息，而且还会给模型带来域分布偏移（因为自然图像中不存在此类标记）并导致性能下降。为了缓解这个问题，作者提出了一种称为掩码提示调整的技术，即视觉提示调整 [12]。

具体来说，当图像输入 CLIP 时，掩膜图像记为张量 $T \in R^{N_p \times E}$ ，其中 N_p 是补丁的数量， E 是标记维度。掩膜图像还带有一个压缩的二进制掩膜 $M_p \in \{0, 1\}^{N_p}$ ，其中每个元素指示给定的补丁是被保留还是被遮掩。仅当补丁内的所有像素都被完全遮掩时，补丁才会被视为遮掩标记。直觉上，边界像素通常存在于部分遮蔽的块中，对于区域分类至关重要。分配一个表示提示标记的可学习张量 $P \in R^{N_p \times E}$ 。最后，transformer 的最终输入为 $T \oplus M_p + P \oplus (1 - M_p)$ ，其中 \oplus 表示逐元素乘法。按照 [12] 中的“深层提示”，我们可以将此类提示标记添加到 transformer 的更深层。图 3 也对此进行了说明。

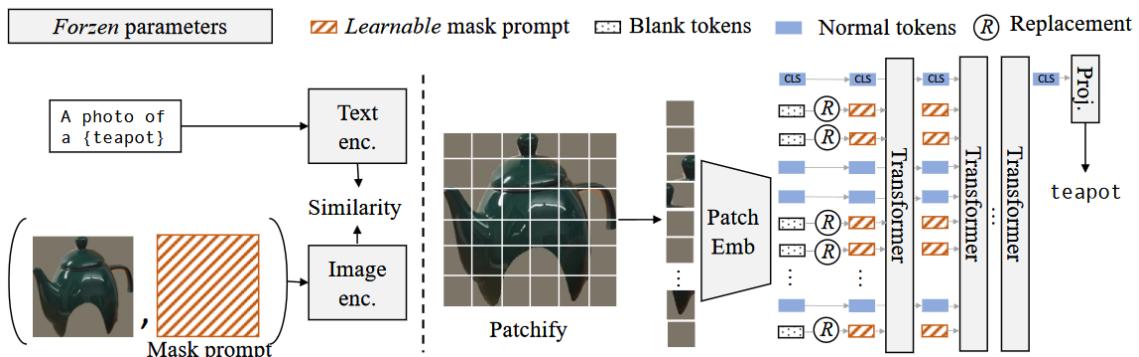


图 3. 掩码提示调整可以使 CLIP 适应掩膜图像而不改变其权重，做法是用可学习的掩码提示替换掩码块中的零标记。

4 复现细节

4.1 与已有开源代码对比

此篇文章的代码在 GitHub 中进行了开源(<https://jeff-liangf.github.io/projects/ovseg>)，在已经开源的代码上，我为了使模型能够适应更细粒度的类别进行语义分割，对现有的分割模块 MaskFormer 使用 SAM 模型进行了一个替换，并对 SAM 分割过于细粒度的情况，设计了一个简单的融合模块进行适应。最终，使用 SAM 进行分割的结果比用 MaskFormer 分割得到的结果更好，且能通过用户的参数调整，进而控制分割结果的细粒程度。

4.2 实验环境搭建

本机的配置如下：Linux 系统、55GB 内存、4 块 NVIDIA Titan X (12GB) 显卡。根据作者的 Readme (<https://github.com/facebookresearch/ov-seg/blob/main/README.md>) 文档进行了环境搭建，主要使 python 库的配置以及所用到的 5 个数据集的下载以及数据集目录的调整。

4.3 复现结果对比

在已经开源的代码上，我先是通过对开源的代码进行了学习，然后在自己的服务器上进行部分训练以及在 ADE20K-150/847、PASCALVOC-20、PASCALContext-59/459 这 5 个数据集上进行测试，看在自己机器上的复现情况是否能够符合文章的指标。经过调整，其模型在自己机器上的性能符合预期。复现的各项指标如图 4 所示。

method	backbone	training dataset	A-847	PC-459	A-150	PC-59	PAS-20
<i>Open-vocabulary generalist models</i>							
SPNet [37]	R-101	PASCAL-15	-	-	24.3	18.3	
ZS3Net [4]	R-101	PASCAL-15	-	-	19.4	38.3	
LSeg [23]	R-101	PASCAL-15	-	-	-	47.4	
LSeg+ [16]	R-101	COCO Panoptic	2.5	5.2	13.0	36.0	59.0
SimBaseline [40]	R-101c	COCO-Stuff-156	-	-	15.3	-	74.5
ZegFormer [11]	R-50	COCO-Stuff-156	-	-	16.4	-	80.7
OpenSeg [16]	R-101	COCO Panoptic	4.0	6.5	15.3	36.9	60.0
OVSeg (Ours)	R-101c	COCO-Stuff-156	7.0	10.4	24.0	51.7	89.2
OVSeg (Ours)	R-101c	COCO-Stuff-171	7.1	11.0	24.8	53.3	92.6
<i>Supervised specialist models</i>							
FCN [29]	Eff-B7	COCO Panoptic	3.8	7.8	18.0	46.5	-
Deeplab [6]	Eff-B7	COCO Panoptic	6.3	9.0	21.1	42.1	-
SelfTrain [45]	Eff-L2	Same as test	-	-	-	-	90.0
MaskFormer [9]	R-101c	Same as test	17.4	-	46.0	-	-

```

[12/08 13:05:39] d2.engine.defaults INFO: Evaluation results for ade20k_full_sem_seg_val in csv format:
[12/08 13:05:39] d2.evaluation.testing INFO: copypaste: Task: sem_seg
[12/08 13:05:39] d2.evaluation.testing INFO: copypaste: mIoU,fwIoU,mACC,pACC
[12/08 20:20:07] d2.engine.defaults: Evaluation results for pascal_context_459_sem_seg_val in csv format:
[12/09 20:20:07] d2.evaluation.testing: copypaste: Task: sem_seg
[12/09 20:20:07] d2.evaluation.testing: copypaste: mIoU,fwIoU,mACC,pACC
[12/09 20:20:07] d2.evaluation.testing: copypaste: [12.2576] 55.5770,24.9899,66.1402
[12/08 09:10:49] d2.engine.defaults INFO: Evaluation results for ade20k_sem_seg_val in csv format:
[12/08 09:10:49] d2.evaluation.testing INFO: copypaste: Task: sem_seg
[12/08 09:10:49] d2.evaluation.testing INFO: copypaste: Task: sem_seg
[12/08 09:10:49] d2.evaluation.testing INFO: copypaste: mIoU,fwIoU,mACC,pACC
[12/08 09:10:49] d2.evaluation.testing INFO: copypaste: [29.6567] 57.2939,48.0265,68.9114
[12/09 05:20:52] d2.engine.defaults INFO: Evaluation results for pascal_context_59_sem_seg_val in csv format:
[12/09 05:20:52] d2.evaluation.testing INFO: copypaste: Task: sem_seg
[12/09 05:20:52] d2.evaluation.testing INFO: copypaste: mIoU,fwIoU,mACC,pACC
[12/09 05:20:52] d2.evaluation.testing INFO: copypaste: [55.7846] 66.6502,75.4969,77.3572
[12/09 13:15:29] d2.engine.defaults INFO: Evaluation results for pascalvoc20_sem_seg_val in csv format:
[12/09 13:15:29] d2.evaluation.testing INFO: copypaste: Task: sem_seg
[12/09 13:15:29] d2.evaluation.testing INFO: copypaste: mIoU,fwIoU,mACC,pACC
[12/09 13:15:29] d2.evaluation.testing INFO: copypaste: [94.7545] 95.0191,97.4911,97.1975

```

图 4. 复现指标对比，评价指标为平均交并比 mIoU

4.4 界面分析与使用说明

为了方便测试以及使用，此项目开发有一个简单的网站 demo，其界面内容如图 5，左侧可以输入想要分割的类别文字，多个类别可用逗号隔开。接下来的二选一按钮则是选择使用哪个分割模型作为分割，如果使用的是 SAM 模型，那么还需要在传入一个细粒度参数，此参数越大，分割的程度越细越精确，但是会出现阈值过高而漏分割的情况。最下面则是选择上传图片的区域。右侧则是效果展示的区域。

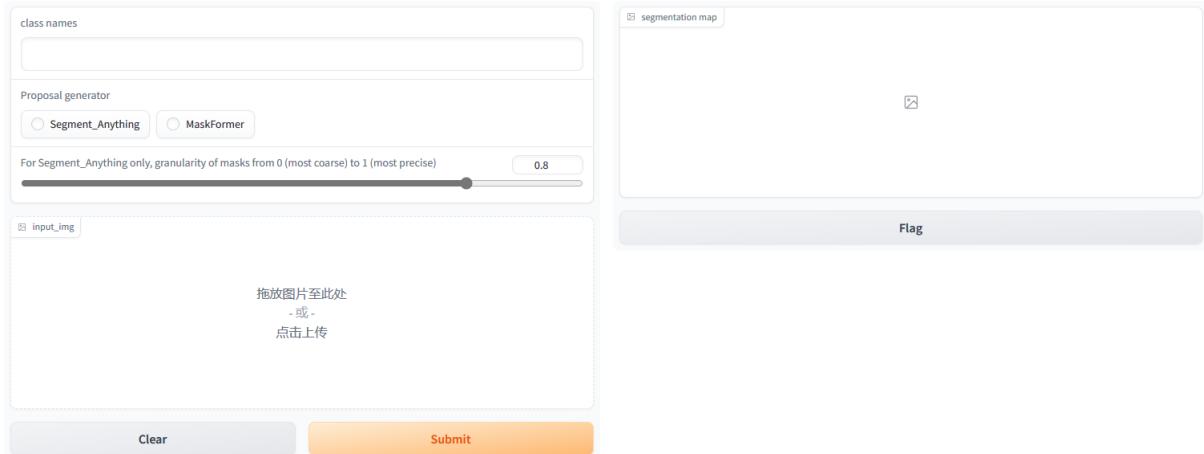


图 5. 操作界面示意

下面是使用这个网站 demo 进行语义分割的一个实例展示，如图 6 所示。

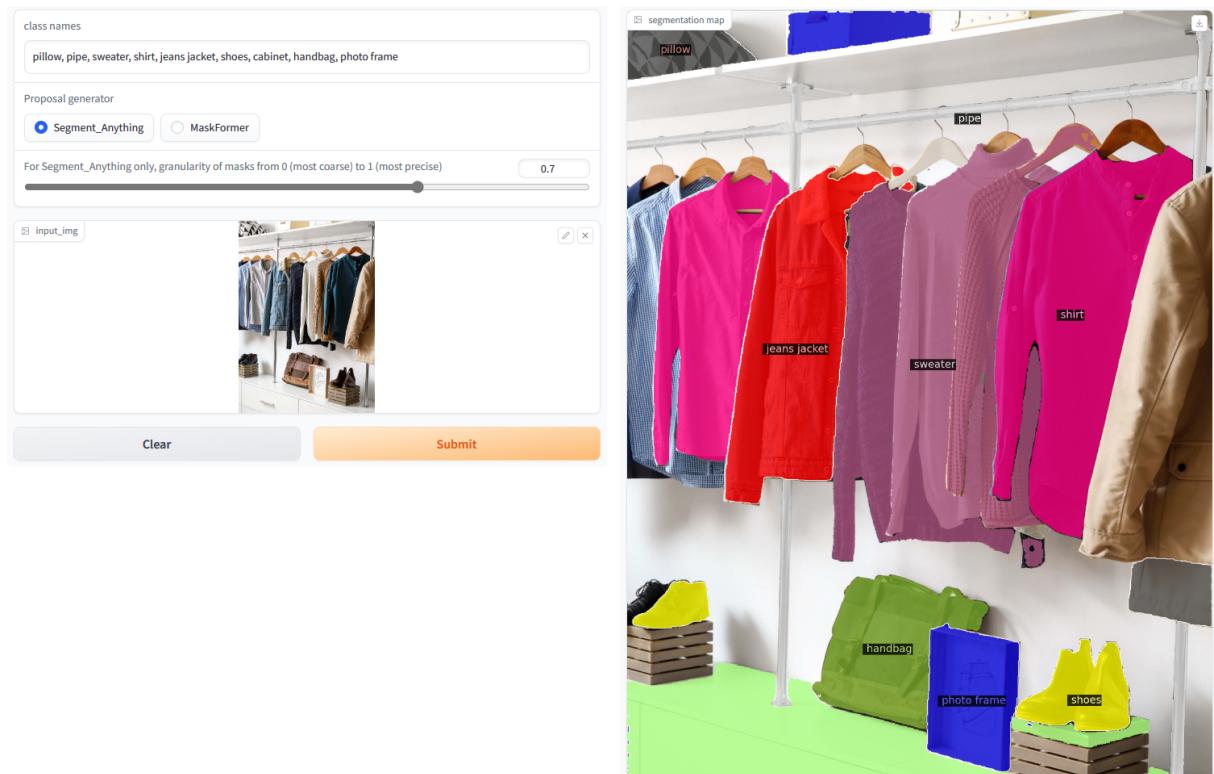


图 6. 操作界面示意

4.5 创新点

在自己使用这个模型进行一些语义分割测试时，发现模型在针对一些不那么细粒度的语义分割时，其性能较好，并且在输入给他的词汇量较少时，其分割效果显著。但是，如果输入的词汇细粒度较细时，并且分割目标较多时，性能急剧下降。模型的分割效果如图 7。

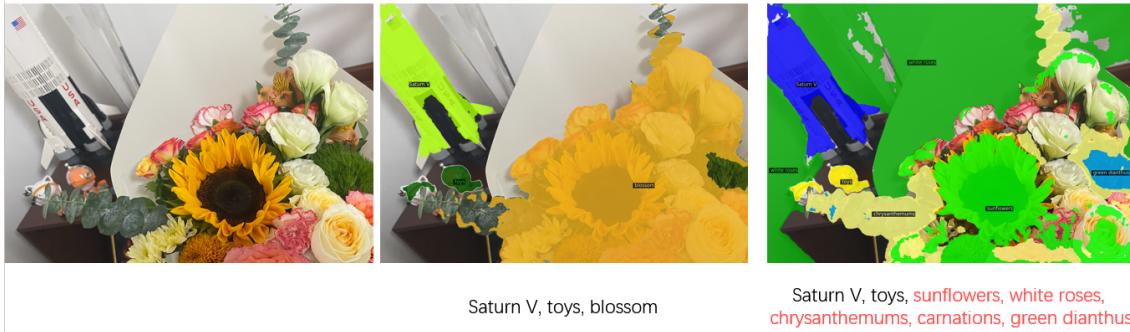


图 7. 使用更加细粒度的类别并增加类别，效果变差

之后我对代码进行了分析，发现分割模型使用的是 MaskFormer，并且这部分模块与后面的 CLIP 分类模块之间的关系是一个正交的关系，即这两个模块相对独立，因此可以根据这个关系，使用其他的模块代替 MaskFormer，以期得到一个更好的语义分割效果。这部分可以使用 2023 年初发表的工作 SAM 替代原来使用的 MaskFormer 模块，因为 SAM 他的性能更好，并且因为使用到的训练数据更加庞大，他的泛化能力也比 MaskFormer 更强，理论上是可以使此模型得到更好的开放词汇语义分割效果的。对原模型图 1 的主要改变处为图 8 所示。

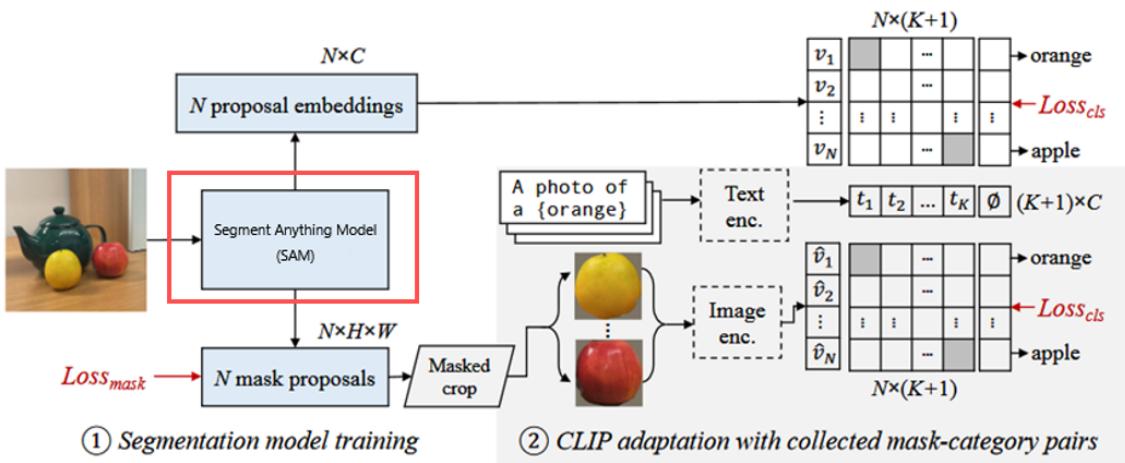


图 8. 与原模型的变动之处为分割模块部分

修改分割模块后，最初试图只保留得分最高的掩码提案，就像在 MaskForm 中一样。然而，这样我们会发现 SAM 得到的掩码是非常细粒度的，比如说：给定一个椅子，它会返回椅子的一条腿或某一椅子的部位，就像图 9，只能得到椅子一部分的掩码。



图 9. SAM 出现的过度分割情况

为了解决 SAM 倾向于过度分割的情况，设置了一个融合模块进行适应，其主要的做法是先把分割模型 SAM 得到的掩码提案全部交给 CLIP 进行分类，之后根据传入的细粒度参数做一个融合。通过设置一个较低的参数值，可以合并各个关于椅子的掩码，如椅子的几个椅腿和其他部分的掩码，最终得到一整个椅子的掩码。当然，通过这种简单的参数设置进行融合是比较直接粗糙的方法，并不能保证得到我们想要的细粒度分割，但是效果也比原来模型的效果要好上不少。此部分的代码实现细节如下：

```
1 # 如果设置的细粒度参数小于1，进行融合，否则直接使用最细粒度的掩码
2 if self.granularity < 1:
3     thr_scores = max_scores * self.granularity # 设置阈值为最大得分乘上细粒度参数值
4     select_mask = [] # 初始化一个空列表，用于保存符合阈值的所有掩码
5
6     # 如果类别名的长度为2且最后一个类别名为'others'
7     if len(class_names) == 2 and class_names[-1] == 'others':
8         thr_scores = thr_scores[:-1] # 去掉最后一个阈值
9
10    # 遍历每一个掩码的得分情况
11    for i, thr in enumerate(thr_scores):
12        cls_pred = class_preds[:, i] # 获取当前阈值对应的类别预测值
13        locs = torch.where(cls_pred > thr) # 找到预测值大于阈值的所有位置
14        select_mask.extend(locs[0].tolist()) # 将这些位置添加到select_mask列表中
```

5 实验结果分析

以下是经过修改后模型的分割效果，首先我与图 7 的结果做了对比，蓝色方框内的数值是得到此结果设置的细粒度参数值，对比结果如图 10。可以发现，使用 SAM 后，无论是分割结果的准确度交并比，还是其细粒度分类的效果，都比原来的 MaskFormer 效果好了很多，尤其是细粒度这方面，原来模型是不能很好进行更细的类别进行识别分割的，后面的例子中也有更多这方面的展示对比。

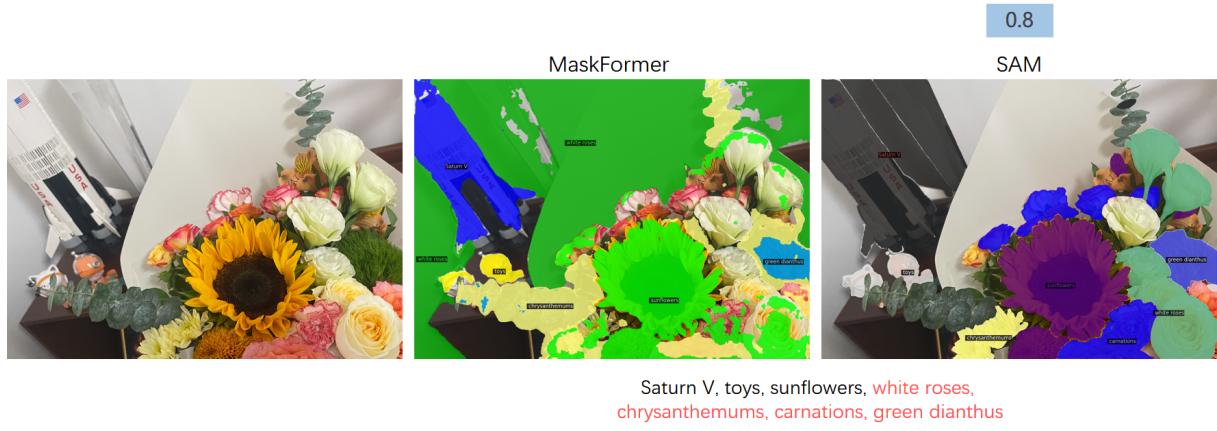


图 10. 效果对比图 1

使用 SAM 后，与原模型相比，它可以对更细粒度的类别进行语义识别和分割，比如下面图 11 中，可以明显看到，虽然我都让模型将 clothes 类别细分为 sweater, shirt, jeans jacket 这三个细粒度的类别，但是可以明显看到，MaskFormer 并不能完成对它们的语义识别，而使用了 SAM 后，则模型可以识别出来并进行一个较好的分类分割。

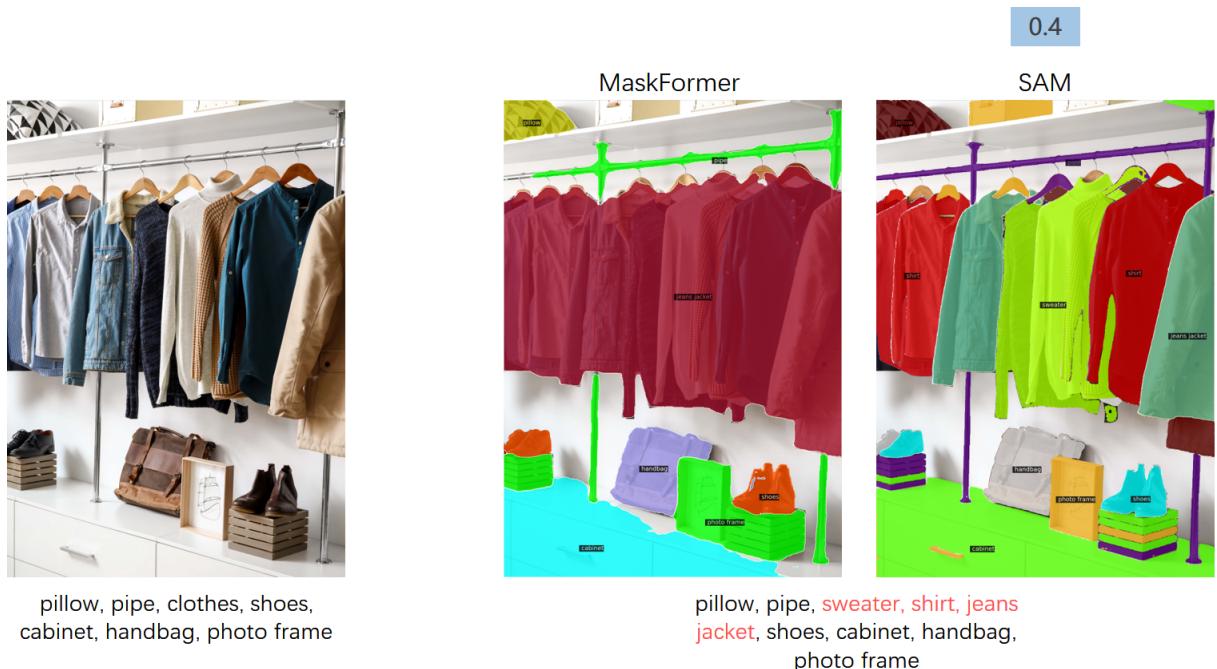
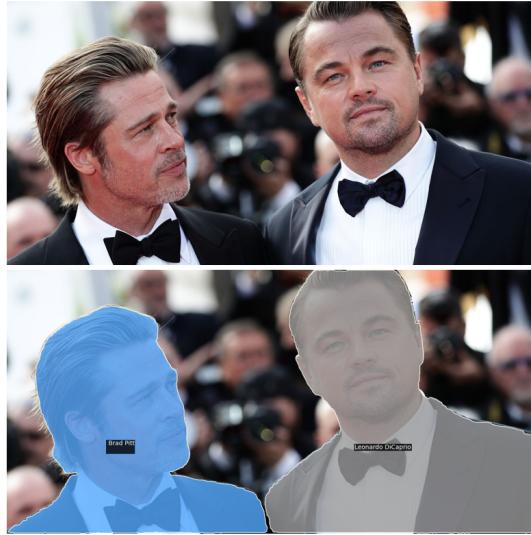


图 11. 效果对比图 2

此外，我还使用了具体的名人来让模型进行一个语义分割，如图 12，也可以看到经过改进后的模型也得到了不错的效果，可以精确认识到我们给他的文本语义信息并对图像进行分割。



LeBron James, Stephen Curry



Leonardo DiCaprio, Brad Pitt

图 12. 改进模型对人物的语义分割效果

6 总结与展望

我经过对这篇优秀论文的复现后，感到目前开放词汇这个新兴的方向很有潜力，目前正在得到越来越多人的关注和推进，这是一个比较前沿的技术。我在复现的同时，也学习了解到了很多这些方向上的技术知识，并且也增进了我的代码能力。目前这个改进是作者提出的，我认为这方面还是比较简单直接的，有以下几个不足之处：1. 首先是在使用了 SAM 模型之后，为解决 SAM 过度分割的情况，额外设置了一个融合模块，模型需要更多的输入，不如之前那么简单直接。2. 在使用融合模块时，需要手动输入一个值，要得到一个比较好的结果只能不断进行尝试，不够自动化。那根据这两个缺点，未来可以将这部分模块进行一个改进，看是否可以将这个细粒度参数也进行与一个学习，让模型能够根据输入的图像自动确定一个效果比较好的值，而不是需要人为判断设置。

参考文献

- [1] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 2022.
- [2] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* ” O'Reilly Media, Inc.”, 2009.
- [3] Maxime Bucher, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Zero-shot semantic segmentation. *Advances in Neural Information Processing Systems*, 32, 2019.
- [4] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

- [5] Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. Vqgan-clip: Open domain image generation and editing with natural language guidance. In *European Conference on Computer Vision*, pages 88–105. Springer, 2022.
- [6] Jian Ding, Nan Xue, Guisong Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. 2022 ieee. In *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11573–11582, 2021.
- [7] Zheng Ding, Jieke Wang, and Zhuowen Tu. Open-vocabulary panoptic segmentation with maskclip. *arXiv preprint arXiv:2208.08984*, 2022.
- [8] Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. Learning to prompt for open-vocabulary object detection with vision-language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14084–14093, 2022.
- [9] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *European Conference on Computer Vision*, pages 540–557. Springer, 2022.
- [10] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021.
- [11] Chao Jia, Yinfai Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021.
- [12] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pages 709–727. Springer, 2022.
- [13] Kwanyoung Kim, Yujin Oh, and Jong Chul Ye. Zegot: Zero-shot segmentation through optimal transport of text prompts. *arXiv preprint arXiv:2301.12171*, 2023.
- [14] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.
- [15] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations*, 2022.
- [16] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

- [17] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. *arXiv preprint arXiv:2110.05208*, 2021.
- [18] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7061–7070, 2023.
- [19] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.
- [20] Huaishao Luo, Junwei Bao, Youzheng Wu, Xiaodong He, and Tianrui Li. Segclip: Patch aggregation with learnable centers for open-vocabulary semantic segmentation. In *International Conference on Machine Learning*, pages 23033–23044. PMLR, 2023.
- [21] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- [23] Bichen Wu, Ruizhe Cheng, Peizhao Zhang, Peter Vajda, and Joseph E Gonzalez. Data efficient language-supervised zero-shot recognition with optimal transport distillation. *arXiv preprint arXiv:2112.09445*, 2021.
- [24] Yongqin Xian, Subhabrata Choudhury, Yang He, Bernt Schiele, and Zeynep Akata. Semantic projection network for zero-and few-label semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8256–8265, 2019.
- [25] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18134–18144, 2022.
- [26] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. Side adapter network for open-vocabulary semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2945–2954, 2023.
- [27] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for zeroshot semantic segmentation with pre-trained vision-language model. *arXiv preprint arXiv:2112.14757*, 3, 2021.

- [28] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16793–16803, 2022.
- [29] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.