A Simple Baseline for Open-Vocabulary Semantic Segmentation with Pre-trained Vision-language Model

Mengde Xu[1,3*], Zheng Zhang[1,3*], Fangyun Wei[3*], Yutong Lin[2,3], Yue Cao[3], Han Hu[3], and Xiang Bai[1†]

[1] Huazhong University of Science and Technology {mdxu,xbai}@hust.edu.cn
[2] Xi'an Jiaotong University yutonglin@stu.xjtu.edu.cn
[3] Microsoft Research Asia {zhez,fawe,hanhu}@microsoft.com,caoyue10@gmail.com

Abstract. Recently, open-vocabulary image classification by vision language pre-training has demonstrated incredible achievements, that the model can classify arbitrary categories without seeing additional annotated images of that category. However, it is still unclear how to make the open-vocabulary recognition work well on broader vision problems. This paper targets open-vocabulary semantic segmentation by building it on an off-the-shelf pre-trained vision-language model, i.e., CLIP. However, semantic segmentation and the CLIP model perform on different visual granularity, that semantic segmentation processes on pixels while CLIP performs on images. To remedy the discrepancy in processing granularity, we refuse the use of the prevalent one-stage FCN based framework, and advocate a two-stage semantic segmentation framework, with the first stage extracting generalizable mask proposals and the second stage leveraging an image based CLIP model to perform open-vocabulary classification on the masked image crops which are generated in the first stage. Our experimental results show that this two-stage framework can achieve superior performance than FCN when trained only on COCO Stuff dataset and evaluated on other datasets without fine-tuning. Moreover, this simple framework also surpasses previous state-of-the-arts of zero-shot semantic segmentation by a large margin: +29.5 hIoU on the Pascal VOC 2012 dataset, and +8.9 hIoU on the COCO Stuff dataset. With its simplicity and strong performance, we hope this framework to serve as a baseline to facilitate future research. The code are made publicly available at https://github.com/MendelXu/zsseg.baseline.

1 Introduction

Semantic segmentation is a fundamental computer vision task that assigns every pixel of an image with category labels. Accompanied by the development of deep learning [27, 42, 23, 13, 35], the semantic segmentation has also evolved tremendously under the supervised learning paradigm [36, 7, 3]. However, unlike

* Equal contribution.
† Corresponding author

common image-level datasets such as ImageNet-1K/ImageNet-22K image classification which are easily scaled up to tens of thousands of categories, existing semantic segmentation tasks involve usually up to tens or hundreds of categories due to the significantly higher annotation cost, and thus limit the segmentors' capability in handling rich semantics.

Zero-shot semantic segmentation [5] is an attempt to break the bottleneck of limited categories. However, the narrowly defined zero-shot semantic segmentation usually only takes a small amount of labeled segmentation data and refuses to make use of any other data/information, consequently resulting in poor performance. In this work, we focus on another more practical setting: *open-vocabulary* semantic segmentation, as a generalized zero-shot semantic segmentation, concentrates more on establishing a feasible method to segment arbitrary classes and allows the use of additional data/information except the segmentation data. Specifically, we propose to leverage a recent advance of image-level vision-language learning model, i.e., CLIP [41].

While the vision-language learning model has learnt a strong vision-category alignment model using rich image-caption data, how to effectively transfer its image-level recognition capability to pixel-level is unclear. An natural idea is to integrate the vision-language model with a fully convolutional networks (FCN) [36], an architecture widely used for fully supervised semantic segmentation. A main difficulty of the integration is that the CLIP model is learnt at image-level, which differs from the granularity of FCN that models semantic segmentation as a pixel classification problem, where a linear classifier is applied on each pixel feature to produce the classification results, with each column of the linear classifier weight matrix representing each category. Empirically, we found the granularity inconsistency lead unsatisfactory performance.

To better leverage the strong vision-category correspondence capability involved in the image-level CLIP model, we pursue mask proposal based semantic segmentation approaches such as MaskFormer [9], which first extracts a set of class-agnostic mask proposals and then classifies each mask proposal into a different category. This two-stage approach decouples the semantic segmentation task into two sub-tasks of class-agnostic mask generation and mask category classification. Both sub-tasks prove well adaptation to handle unseen classes: firstly, the class-agnostic mask proposal generation trained using *seen* classes is observed well generalizable to *unseen* classes; secondly, the second mask proposal classification stage is at a same recognition granularity than that used in a CLIP model. To further bridge the gap with a CLIP model, the masked image crop of each proposal is used as input to the CLIP model for *unseen* classes classification. In addition, we employ a prompt-learning approach [34] to further improve the *unseen* classes classification accuracy given a pre-trained CLIP model.

We evaluate the proposed approach under two different settings: 1) *Cross-dataset* setting where the model is trained on one dataset and evaluated on other datasets without fine-tuning. Under this setting, our two-stage framework demonstrate well generalization capability. It outperforms FCN approach by **+13.1** mIoU on Cityscapes, **+19.6** mIoU on Pascal Context, **+5.6** mIoU on

ADE20k with 150 classes and **+2.9** mIoU on ADE20k with 847 classes. 2) *Zero-shot* setting where the model is trained on a part of *seen* class of a dataset and evaluated on all classes (including *seen* and *unseen* classes). We use this setting for comparing with other zero-shot semantic segmentation methods. We show that the proposed approach, though simple and straightforward, can surpass previous state-of-the-arts zero-shot segmentation approaches [5, 40, 48, 20] by a large margin. On Pascal VOC 2012 [14], this approach outperforms previous best methods that w/o self-training by **+37.8** hIoU , and by **+29.5** hIoU when an additional self-training process is involved. On COCO Stuff [6], the approach outperforms previous best methods that w/o self-training by **+19.6** hIoU and by **+8.9** hIoU when an additional self-training process is involved. We hope our simple but effective approach can encourage more study in this direction.

2 Related Works

Vision-Language Pre-training. Vision-language pre-training focuses on how to connect visual concepts and language concepts. Early approaches [44, 37, 8, 31, 33] were performed on some cleaned datasets with relatively small data scale. Therefore, those models usually need to be fine-tuned on some specific downstream tasks. Some recent works [41, 25] have explored the benefits of large-scale noisy data obtained from web pages for vision-language pre-training. CLIP [41], as a representative work, employs a contrastive learning approach to distinguish the correct image-text pair in each training batch. Because many vision/language concepts are covered in large-scale data, the CLIP illustrates surprisingly strong capability on zero-shot/open-vocabulary image classification and image-text retrieval. This work introduces the CLIP model as a strong vision-category correspondant for open-vocabulary semantic segmentation.

Semantic Segmentation. Semantic segmentation is a fundamental task in computer vision that aims to assign a category to each pixel. Fully convolutional network [36] and its variants [7, 49, 3], as a practical and straightforward approach to model the semantic segmentation as a pixel-wise classification problem, have dominated this field in the past few years. Recently, MaskFormer [9] explored to model the semantic segmentation as two sub-tasks: segment generation and segment classification and has shown competitive performance compared to FCN based approaches.

Zero-Shot Learning and Open-vocabulary Learning. Zero-shot learning has been widely studied in recent years. A narrowly defined zero-shot learning focuses on learning transferable representations from the annotated data of seen classes to represent unseen classes. For example, [1, 47] proposed to learn a joint embedding space between the images and the name/description of the category for image classification, and [28] explored taking the advantages of mid-level semantic representation. Recently, the open-vocabulary learning has attracted

more attentions. As a generalized zero-shot learning, the open-vocabulary learning is more concerned with establishing a feasible method for arbitrary class recognition and allows the use of any additional information. For example, Visual N-Grams [29] and CLIP [41] explored the use of web-crawled data for image classification and [19] introduced the vision-language pre-training model for the open-vocabulary object detection and showed it could significantly improve the long-tile object detection [22].

Zero-shot Semantic Segmentation. Some pioneer works to study the zero-shot learning for semantic segmentation. ZS3Net [5] uses generative models to synthesize pixel-level features by word embeddings of *unseen* classes. CSRL [32] further incorporating the structural relation in feature synthesize. CaGNet [20, 21] introduce a contextual module for better feature generation. Different from [5, 32, 20], SPNet [48] attempt to mapping vision feature to the semantic space via word embedding. JoEm [4] a joint embedding strategy between the vision encoder and semantic encoder. In [26], variational mapping is used to learn semantic features. In [24], the uncertainty-aware losses are proposed to eliminate noisy samples. Other works explored other directions or aspects of zero-shot semantic segmentation. In [11], the super-pixel pooling is utilized to improve the region grouping generalization. In [40], the self-training for zero-shot semantic segmentation are carefully studied. In [43, 38], the transductive learning setting are explored. In [45] , they explore the utilization of image caption. However, all those methods have not explored the utilization of the vision-language pre-training model in zero-shot semantic segmentation. There are two concurrent work [30, 16] try to utilize the vision-language pre-training model in semantic segmentation. However, LSeg [30] is an FCN-based approach focus on few shot setting. Openseg [16], which is a similar work to ours, utilizes external grounding dataset while we don't. In addition, Openseg is based on ALIGN [25] while we adopt CLIP [41].

3 Preliminary

In this section, we first introduce the setting of open-vocabulary semantic segmentation and revisit CLIP as preliminary.

3.1 Open-vocabulary Semantic Segmentation

Zero-Shot Setting. Open-vocabulary is an generalized zero-shot task, so the zero-shot semantic segmentation protocol can also evaluate open-vocabulary semantic segmentation. In this setting, model predicts masks for *unseen* classes $\mathcal{C}^{\text{unseen}}$ by learning from some labeled data of *seen* classes $\mathcal{C}^{\text{seen}}$, and the *seen* classes and *unseen* classes are disjoint, i.e., $\mathcal{C}^{\text{unseen}} \cap \mathcal{C}^{\text{seen}} = \varnothing$. Usually, $\mathcal{C}^{\text{seen}}$ and $\mathcal{C}^{\text{unseen}}$ are often represented with semantic words like *dog, cat, apple*, and sometimes the description of the classes are also provided.

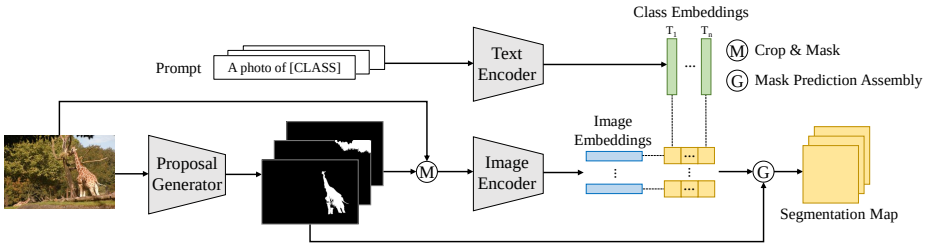

Fig. 1. Overview of our two-stage open-vocabulary semantic segmentation framework. We reformulate and break down the open-vocabulary semantic segmentation into two steps: 1) training a mask proposal generator to generate a set of binary masks; 2) leveraging the pre-trained CLIP to classify each mask proposal.

During training, a training set $\mathcal{X}_{\text{train}} = \{(\mathcal{I}_k, \mathcal{M}_k)\}$ with input images $\mathcal{I}_k$ and the ground-truth semantic segmentation annotations $\mathcal{M}_k$ is provided, and the training annotations $\{\mathcal{M}_k\}$ contains only the *seen* classes. The trained model is evaluated on a testing set $\mathcal{X}_{\text{test}}$, both *seen* classes and *unseen* classes need to be predicted in testing set $\mathcal{X}_{\text{test}}$.

Cross-Dataset Setting. In this setting, the model is trained on one dataset and evaluated on another dataset without fine-tuning. This is a more challenging setting than the *zero-shot setting*, where the model not only deals with the *unseen* classes, but also has to address the domain gap among different datasets.

3.2 Revisiting CLIP

CLIP [41] is a powerful pre-trained vision-language model, which shows surprisingly strong performance in associating the visual and textual concepts. CLIP is a two-stream method: it contains an image encoder $\mathcal{E}_{\text{image}}$ and a text encoder $\mathcal{E}_{\text{text}}$. For any given image-text paired data $\{\mathcal{I}, \mathcal{T}\}$, their semantic similarity can be estimated by computing the cosine distance between $\mathcal{E}_{\text{image}}(\mathcal{I})$ and $\mathcal{E}_{\text{text}}(\mathcal{T})$.

The pre-trained CLIP model can be used to classify images by a given set of classes without fine-tuning, which is also known as zero-shot/open-vocabulary image classification. Specifically, the class names are injected into the pre-defined prompt template and fed into CLIP's text encoder to generate the class embeddings, e.g., a typical prompt template is 'a photo of [CLASS]', where [CLASS] is replaced by the specific class name such as 'person' and 'cat'. The generated class embeddings are used as the classifier and the similarity with image embedding is computed for classification.

In this work, we extend the compatibility of CLIP from *image-level* zero-shot/open-vocabulary classification to *pixel-level* open-vocabulary semantic segmentation, by exploring the use of a pre-trained CLIP model as a strong vision-category correspondent.

4 Two-Stage Open-Vocabulary Semantic Segmentation

Figure. 1 shows an overview of our two-stage framework. Given an image, a set of mask proposals are first generated, and then each proposals is fed into an image encoder and compared with the class weights obtained by applying text encoder on the prompt class description to perform the classification. Finally, the mask prediction are assembled together to produce the final segmentation results. We will describe each component of our framework in the following.

4.1 Mask Proposal Generation

We first introduce the mask proposal generation. In our work, we try three different methods to generate the mask proposals $\{\mathcal{M}_k^p\}$:
GPB-UCM [2]. This is a classical method to generate hierarchical segments by considering multiple low-level cues, e.g., brightness, color, texture, and local gradients. The generated segments of this approach are usually well aligned with the contour of objects.
Selective Search [46]. This method can also generate hierarchical segments. Since this method can effectively localize objects, it is widely used in object detection systems [17, 18].
MaskFormer [9]. This is a recently proposed method for supervised semantic segmentation. Unlike a fully convolution network that models the semantic segmentation as the pixel-wise classification problem, MaskFormer disentangles the semantic segmentation into two sub-tasks: predicting the segments at first and then classifying the category of each segment. We observe that the predicted segments by MaskFormer can be used as the mask proposals, and we empirically demonstrate (see Table. 6) that the MaskFormer trained on *seen* classes can produce high-quality mask proposals on the *unseen* classes. Therefore, we take this advantage of MaskFormer as our default mask proposal generator.

4.2 Region Classification via CLIP

Two Strategies for Using CLIP. There are two strategies to perform the region classification by utilizing the pre-trained CLIP:

- The first strategy is to directly apply the CLIP image encoder on each mask proposals for classification. Specifically, given an image $\mathcal{I}$ and a mask proposal $\mathcal{M}^p$, the mask proposals are first binarized with a threshold of 0.5, and then apply the binarized $\mathcal{M}^p$ to image $\mathcal{I}$, erase the unused background and only crop foreground area. The masked image crop is resized to 224^2 and then fed into CLIP for classification. However, since there is no extra training process, the training data of *seen* classes cannot be utilized, resulting in inferior performance on *seen* classes in the inference (see Table. 7).
- To utilize the training data of *seen* classes, another approach is to retrain an image encoder. However, if we simply learn a set of new classifiers on the

training data of *seen* classes, the retrained image encoder has no generalization ability on *unseen* classes since these classes have no corresponding classifiers. Therefore, we propose to use the features generated from the text encoder of the pre-trained CLIP model as the fixed classifier weights for the retrained image encoder. In this approach, the image encoder has a certain generalization ability to the *unseen* classes since the image encoder is encouraged to embed the vision features into the same embedding space of the text encoder through the *seen* classes. Notably, this approach can be easily integrated into the training process of the MaskFormer, by simply using the CLIP generated text features as the classifier weights of the MaskFormer, thus avoiding the need of training an additional image encoder.

The two strategies complement each other (see Table. 7), therefore we ensemble the results of these two strategies by default. Given a mask proposal $\mathcal{M}^p$, we crop the foreground area $A_{fg} = \text{crop}(\mathcal{M}^p, \mathcal{I})$ (See Appendix for details), and compute its classification probability via CLIP vision encoder E_{vision} and text encoder E_{text}:

$$C_i(A_{fg}) = \frac{\exp(\text{cosine}(E_{\text{vision}}(A_{fg})), E_{\text{text}}(\mathcal{C}_i)/\tau)}{\sum_i^{\#\text{class}} \exp(\text{cosine}(E_{\text{vision}}(A_{fg}), E_{\text{text}}(\mathcal{C}_i))/\tau)} \tag{1}$$

,where $\mathcal{C}_i$ is name of i-th class and temperature τ=100. The classification probability of CLIP can be ensembled with supervised model trained on seen classes and then generate final mask results according to Sec.4.3.

Prompt Design. The original CLIP is not designed for open-vocabulary semantic segmentation. How to design feasible text prompts need to be explored.

Hand-Crafted Prompt. A simple approach is to re-use the hand-crafted prompts provided by CLIP which is originally designed for image classification on ImageNet-1K [12]. There are 80 different prompts, each consisting of a natural sentence with a blank position for injecting the category names. Since these prompts are not originally designed for semantic segmentation, some of them may have a adverse effect. So we evaluate each of these prompts on training data to select one most helpful prompt for open-vocabulary semantic segmentation.

Learning-Based Prompt. Prompt learning [34, 51] recently showed great potential for adapting the pre-trained language/vision-language models on specific downstream tasks. We also explore this technique. Specifically, a prompt is a sequence of tokens. Each token belongs to one of the two types: $[P]$ indicates the prompt token and $[CLS]$ indicates the class token. A generalized prompt can be formulated as $[P]_0...[P]_m[CLS]$, where m is the number of prompt token. In prompt learning, the prompt tokens $[P]_0...[P]_m$ are set as learnable parameters that can be trained on the *seen* classes and generalized to the *unseen* classes.

4.3 Mask Prediction Assembly

Since the mask proposals may overlap each other, resulting in the possibility of some pixels being covered by several different mask proposals. Therefore, we employ a simple aggregation mechanism to generate semantic segmentation results from the mask predictions. Specifically, for a given pixel q, its predicted probability of being i-th category is defined as:

$$C_i(q) = \frac{\mathcal{M}_k^p(q) C_k^p(i)}{\sum \mathcal{M}_k^p(q)}, \tag{2}$$

where $\mathcal{M}_k^p(q)$ denotes the predicted probability of pixel q in k-th mask proposal $\mathcal{M}_k^p$, and $C_k^p(i)$ is the predicted probability of mask proposals $\mathcal{M}_k^p$ belonging to i-th category. Note that the sum of $C_i(q)$ over all categories is not guaranteed to be 1, and pixel q is classified to the category with highest predicted value.

5 Fully Convolution Network Approach

In addition to our proposed two-stage framework, a more conventional approach is to use the widely-used fully convolution network (FCN). As a dominant method in supervised semantic segmentation, FCN formulates the semantic segmentation as a pixel-wise classification problem. Specifically, given an image, FCN generates a high-resolution feature map, and a set of learned classifiers is applied on each pixel to produce segmentation predictions. Similar to our proposed two-stage framework, there are also two strategies to apply the CLIP on FCN framework:

- Directly using the feature map generated by the CLIP vision encoder to perform pixel-wise classification. Note that in the original CLIP model, the feature of an image are represented by the feature of `[CLS]` token, not the feature map, and this difference may lead to performance degradation. In addition, the original CLIP model uses the image size of 224×224 during pre-training, while semantic segmentation usually requires a higher image resolution (e.g., shorter size is 640). Therefore, the direct use of high-resolution image during inference may lead to inferior performance due to inconsistency in image size. To alleviate this problem, we try to use the sliding window technique, which is widely used in previous works [7] for performing multi-scale inference. We empirically found that it can improve performance and thus use it by default.
- The training data of the *seen* classes cannot be utilized in the first strategy. Instead, we retrain an FCN-based vision encoder on *seen* classes via the similar method introduced in Sec. 4.2. Specifically, we use the CLIP text encoder to generate a fixed classifier weight. Therefore, the retrained model can obtain a certain generalization ability to the *unseen* classes.

As the same as the two-stage framework, we also ensemble the prediction of these two strategies by default if not specified.

6 Experiments

6.1 Dataset and Evaluation Protocol

Dataset. We conduct extensive experiments on five challenging datasets to evaluate our method: COCO Stuff [6], Pascal VOC 2012 [14], Pascal Context [39], Cityscapes [10], and ADE20K [50].
COCO Stuff is a large-scale dataset that contains 117k training images and 5k validation images. It contains annotations of 171 classes, 80 thing classes and 91 sutff classes respectively.
Pascal VOC 2012 contains 11,185 training images and 1,449 validation images from 20 classes. The provided augmented annotations are used.
Cityscapes is a scene parsing dataset collected on urban streets, containing 5,000 finely annotated images and 20,000 coarsely annotated images. According to the common practices [10], we use 1,525 images of 19 classes in the finely annotated set for validation.
Pascal Context is an extensive dataset of Pascal VOC 2010, containing 4,998 training images and 5,005 validation images. We use the frequent 59 classes for validation.
ADE20K contains 20k training images, 2k validation images, and 3k testing images. There are two settings of 150 classes and 857 classes.

Data Split. For *Cross-dataset setting*, we train our model on the COCO Stuff dataset and test on the validation set of the others. For *Zero-shot setting*, we evaluate our method on COCO Stuff and Pascal VOC 2012. Following [48], we divide the COCO Stuff dataset into 156 *seen* classes and 15 *unseen* classes and the Pascal VOC 2012 dataset into 15 *seen* classes and 5 *unseen* classes.

Evaluation Protocol. For *cross-dataset setting*, we use the mean of class-wise intersection over union (mIoU) as major metric. For *zero-shot setting*, we use harmonic mean IoU (hIoU) among the *seen* classes and *unseen* classes as major metric by following previous works [48, 40] (see Appendix for detail definition). We also report the pixel-wise classification accuracy (pAcc) as a reference.

6.2 Implementation Details

We conduct all experiments on 8×Nvidia V100 GPUs. We train a MaskFormer [9] model on the COCO Stuff dataset with ResNet-101 as the default backbone. An AdamW optimizer with the initial learning rate of 1e-4, weight decay of 1e-4 and a backbone multiplier of 0.1, and a poly learning rate policy with a power of 0.9 are used. The batch size is set to 32 for each GPU, and the total training iteration is 60K/120K for zero-shot setting and cross-dataset setting,respectively. If not specified, the MaskFormer model is only trained on *seen* classes, and we use 100 mask proposals for both training and testing. For all other settings and hyper-parameters, we keep the original setting of MaskFormer without changes.

Table 1. We train our model on COCO Stuff dataset and evaluate on other datasets (cross-dataset setting). The number in the parentheses after the dataset name represents class number. Both methods are tuned through the same prompt engineering [19].

Method \ Dataset	Cityscapes (19)	Pascal Context (59)	ADE20K (150)	ADE20K (847)
FCN	21.4	28.2	14.9	4.1
Ours	34.5	47.7	20.5	7.0

CLIP with ViT-B/16 backbone is used by default if not specified. In text prompt tuning, the prompts are randomly initialized, and a SGD optimizer is used to train the learnable prompts. The learning rate is set to 0.02 and decayed according to the cosine learning rate policy, and the batch size is set to 32. We train 50 and 100 epochs for Pascal VOC and COCO Stuff, respectively. For Pascal VOC 2012 dataset, we use a batch size of 16 and a total training iteration of 20K, and keep all other setting as the same as the COCO Stuff dataset.

Table 2. Comparison with other methods on COCO Stuff in the zero-shot setting.

Method	hIoU	mIoU	
		seen	unseen
SPNet [48]	16.8	20.5	14.3
ZS3 [5]	15.0	34.7	9.5
CaGNet [20]	18.2	35.5	12.2
FCN	20.9	30.1	16.0
Ours	37.8	39.3	36.3
SPNet+ST [48]	30.3	34.6	26.9
ZS5 [5]	16.2	34.9	10.6
CaGNet+ST [20]	19.5	35.6	13.4
STRICT [40]	32.6	35.3	30.3
Ours+ST	**41.5**	**39.6**	**43.6**

Table 3. Comparison with other methods on Pascal VOC in the zero-shot setting.

Method	hIoU	mIoU	
		seen	unseen
SPNet [48]	25.1	73.3	15.0
ZS3 [5]	28.7	77.3	17.7
CaGNet [20]	39.7	78.4	25.6
FCN	50.7	85.5	36.0
Ours	77.5	**83.5**	72.5
SPNet+ST [48]	38.8	77.80	25.8
ZS5 [5]	33.3	78	21.2
CaGNet+ST [20]	43.7	78.6	30.3
STRICT [40]	49.8	82.7	35.6
Ours+ST	**79.3**	79.2	**78.1**

6.3 Comparison in Cross-Dataset Setting

We first evaluate our method on the cross-dataset setting. The model is trained on the COCO Stuff dataset and then evaluated on other datasets without fine-tuning. Table. 1 clearly shows that our two-stage approach outperforms the FCN approach by a noticeable margin, demonstrating that our two-stage approach can better leverage the pre-trained CLIP model than the FCN approach. We do not list the result on Pascal VOC as its categories overlap much with the COCO Stuff dataset, and our method can achieve 88.4 mIoU.

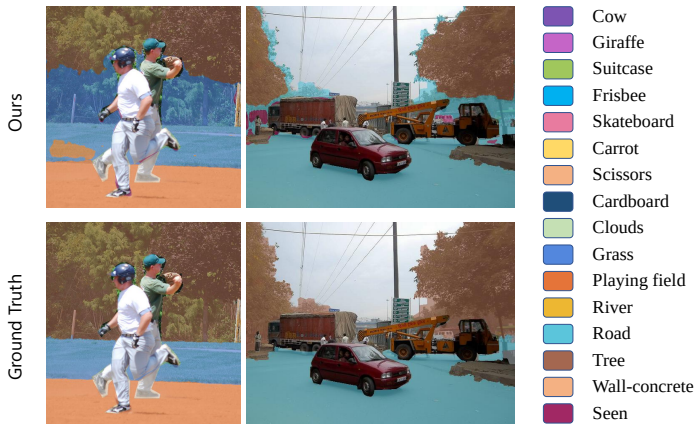

Fig. 2. Qualitative results on COCO Stuff dataset. Only results of unseen classes are visualized. Predictions misclassified to seen classes are labeled with *Seen* color.

6.4 Comparison in Zero-Shot Setting

We then compare our method with previous state-of-the-arts on Pascal VOC 2012 dataset and COCO Stuff dataset. Since some works reported the performance by applying the self-training techniques (denoted as "ST"), we follow this practice and report the performance with or without self-training.

COCO Stuff. Table. 2 shows the results. Compared with Pascal VOC 2012 dataset, COCO Stuff is more challenging. However, our approach still outperforms state-of-the-arts by a large margin. Specifically, without using the self-training, our method achieves 37.8 hIoU and 36.3 mIoU-unseen, outperforming the previous best method CaGNet [20] by +19.5 hIoU and +24.1 mIoU-unseen. By further employing the self-training, our method achieves 41.5 hIoU and 43.6 mIoU-unseen, outperforming the previous best method STRICT [40] by +8.9 hIoU and +13.3 mIoU-unseen. The qualitative results are shown in Figure. 2.

Pascal VOC 2012. The results are shown in Table. 3. Without using the self-training, our method achieves 77.5 hIoU and 72.5 mIoU-unseen, outperforming the previous best method CaGNet [20] by a huge margin of +37.7 hIoU and +46.8 mIoU-unseen. By further employing the self-training, our method achieves 79.3 hIoU and 78.1 mIoU-unseen, outperforming the previous best method STRICT [40] by +29.5 hIoU and +42.5 mIoU-unseen.

While our method outperforms other state-of-the-art zero-shot semantic segmentation methods, **how the larger pre-trained data and image encoder affects the performance is still unclear**. To study these impacts, we design a new implementation that enables our approach to only leverage ImageNet-1K classification data. Specifically, we train a vision-language model by only using ImageNet-1K: the class names of ImageNet-1K are treated as language inputs,

Table 4. Study on how image encoder and pre-trained data affects the performance on Pascal VOC 2012 in the zero-shot setting.

Method	Image Encoder	Pre-train Data	hIoU	mIoU	
				seen	unseen
ZS3 [5]	ResNet-101	ImageNet	28.7	77.3	17.7
CaGNet [20]	ResNet-101	ImageNet	39.7	78.4	25.6
SPNet [48]	ResNet-101	ImageNet	25.1	73.3	15.0
	ResNet-101	CLIP-VL	33.4	74.1	21.5
Ours	ResNet-101	ImageNet	49.5	71.1	38.0
	ResNet-101	CLIP-VL	74.2	**84.6**	66.1
	VIT/B-16	CLIP-VL	**77.5**	83.5	**72.5**

Table 5. Study of different proposal generation methods on COCO Stuff dataset.

Method	hIoU	pAcc	mIoU-unseen
GPB-UCM [2]	10.9	9.5	11.6
Sel. Search [46]	11.0	23.5	13.3
MaskFormer [9]	**28.2**	**48.4**	**29.7**

Table 6. Evaluate the generalization ability of mask proposal generator.

Training set	Test set	mIoU	pAcc
COCO Stuff	COCO Stuff	69.4	87.7
ADE20K	COCO Stuff	62.5	84.6
ADE20K	ADE20K	71.6	90.2
COCO Stuff	ADE20K	64.4	87.7

and are encoded through a pure text encoder[4] to generate the classification weights. As shown in Table. 4, our method achieves 49.5 hIoU with ResNet-101 as backbone, which is much higher than other approaches. On the other hand, we also try to integrate the CLIP with SPNet, and we find it only achieves 33.4 hIoU, which is far from our method by using the same ResNet-101 backbone. Those experiments indicate that the surpassing performance of our method does not only come from larger pre-training data, but also our two-stage framework.

6.5 Ablation Studies

In this section, we validate the key designs of our method. If not specified, we report the performance on the COCO Stuff dataset with the MaskFormer model of ResNet-101 and CLIP of ViT-B/16 by using the *zero-shot setting*.

Different Mask Proposal Generation Methods. We evaluate the performance of the mask proposal generation methods by plugging them into our pipeline. To avoid the impact of the learnable classifier trained on *seen* classes, we perform the comparison by directly classifying the masked regions with the CLIP model. The results are shown in Table. 5, and the MaskFormer achieves better performance than the Selective Search and GPB-UCM. Note that even the other two methods are worse than the MaskFormer, they still achieve comparable performance compared with state-of-the-arts on mIoU-unseen.

[4] We use SimCSE [15] as the text encoder trained on text data only.

Table 7. Study of different region classification methods on COCO Stuff dataset.

Method	hIoU	mIoU	
		seen	unseen
Retrained Vision Enc.	8.7	38.7	4.9
CLIP Vision Enc.	28.2	26.8	29.7
Ensemble	**37.8**	**39.3**	**36.3**

Table 8. Evaluate the performance of different CLIP variants in our framework on COCO Stuff dataset with manual prompt.

Backbone	hIoU	mIoU-unseen
ResNet-50	15.2	16.0
ResNet-101	13.8	12.5
ViT-B/32	15.3	15.7
ViT-B/16	**18.3**	**19.5**

Table 9. Comparison with supervised method on COCO Stuff validation dataset. Sup: MaskFormer trained on both *seen* classes and *unseen* classes.

Method	hIoU	mIoU	mIoU-unseen
Sup	**49.4**	**42.6**	**62.6**
Ours	37.8	39.2	36.3
Ours+ST	41.5	39.9	43.6

Table 10. Comparison with supervised method on the *unseen* classes of COCO Stuff validation set. Δ is the difference in mIoU between things and stuff of *unseen* classes.

Method	mIoU on *unseen* classes			
	all	thing	stuff	Δ
Sup	**62.6**	**67.8**	**58.3**	9.5
Ours	36.3	44.3	29.5	14.9
Ours+ST	43.6	48.4	39.5	8.9

Generalization of Mask Proposal Generator. Although using MaskFormer to generate the mask proposals achieves excellent performance on zero-shot setting, it is still unknown whether Maskformer can produce good performance on the cross-dataset setting, i.e., training on one and testing on another dataset. Therefore, we evaluate the generalization ability of using MaskFormer to generate mask proposals between the COCO Stuff dataset and the ADE20K dataset.

In this experiment, we want to evaluate only the quality of the proposal without the effects of the region classifier. However, it is difficult to design a simple "recall" metric for mask proposals in semantic segmentation similar to object detection. Because a segment can consist of multiple mask proposals, this may lead low recall while the final semantic segmentation result is still correct. Therefore, we designed an "*oracle*" experiment to evaluate how these proposals affect the final performance of semantic segmentation. Specifically, for each mask proposal, its category is specified as the same as the ground-truth segment in which it has the largest overlap. In this case, the segmentation performance can fully reflect the proposal quality.

The results are shown in Table. 6. We directly report the mIoU in this experiment because the *seen* class cannot be defined between different datasets. We note that the MaskFormer model trained on COCO Stuff can produce good performance on ADE20K compared to the MaskFormer model directly trained on ADE20K with acceptable performance degradation, and vice versa. That demonstrates the generalization ability to use MaskFormer as the proposal generator.

Different Strategies of Using CLIP. We study the two different strategies of using CLIP discussed in Sec. 4.2: retrained vision encoder or directly using

CLIP vision encoder without tuning. The results are shown in Table. 7. The retrained vision encoder shows excellent performance on *seen* classes, while its performance on *unseen* classes is relatively low. In contrast, the CLIP vision encoder shows strong performance on *unseen* classes while worse than retrained vision encoder on *seen* classes by a large margin. By ensembling the two strategies, the performance on both *seen* and *unseen* classes is significantly improved, indicating the two strategies are complementary.

Different CLIP Variants. CLIP provides several variants with different network architectures and model sizes. We study how these models affect the performance of our method when using them as the region classifiers. We report the results without using the learnable prompt due to the high experimental overhead. The results are shown in Table. 8. We find that all models perform well and CLIP with ViT-B/16 achieves the best performance.

Comparison with Supervised Baseline. We also compare our method with the supervised baseline on COCO Stuff. The supervised model is MaskFormer with ResNet-101 backbone which is trained on all classes, including *seen* and *unseen* classes. We report the results in Table. 9. It is remarkable that while our method is worse than the supervised baseline by a large margin on mIoU-unseen and hIoU, the gap in mIoU is much close. That is because there are only 15 *unseen* classes in the current dataset partition. For reference, there are 156 *seen* classes.

To further explore the performance gap between our method and the supervised baseline, we split the *unseen* classes into *things* and *stuff*. The results are reported in Table. 10. We find the performance gap of our method between *things* classes and the *stuff* classes is significantly large than the supervised baseline, and self-training can significantly reduce the gap. This observation suggests that the classification ability of CLIP models is different in *things* and *stuff*, which may be due to the bias of the pre-trained dataset used by CLIP.

7 Conclusion

In this work, we propose a simple and effective two-stage framework for open-vocabulary semantic segmentation with the advanced pre-trained vision-language model. We reformulate and break down the open-vocabulary semantic segmentation into two steps: 1) training a mask proposal generator to generate a set of binary masks and 2) leveraging the pre-trained CLIP to classify each mask proposal. We conduct extensive experiments to verify our approach. Notably, the proposed framework outperforms previous state-of-the-arts of zero-shot semantic segmentation on Pascal VOC 2012 and COCO Stuff by large margins. Our work reveals the potential for using pre-trained vision-language models on open-vocabulary/zero-shot semantic segmentation and provides a strong baseline for this community to facilitate future research.

References

1. Akata, Z., Perronnin, F., Harchaoui, Z., Schmid, C.: Label-embedding for image classification. IEEE transactions on pattern analysis and machine intelligence **38**(7), 1425–1438 (2015)
2. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE transactions on pattern analysis and machine intelligence **33**(5), 898–916 (2010)
3. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence **39**(12), 2481–2495 (2017)
4. Baek, D., Oh, Y., Ham, B.: Exploiting a joint embedding space for generalized zero-shot semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9536–9545 (2021)
5. Bucher, M., Vu, T.H., Cord, M., Pérez, P.: Zero-shot semantic segmentation. Advances in Neural Information Processing Systems **32**, 468–479 (2019)
6. Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1209–1218 (2018)
7. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence **40**(4), 834–848 (2017)
8. Chen, Y.C., Li, L., Yu, L., El Kholy, A., Ahmed, F., Gan, Z., Cheng, Y., Liu, J.: Uniter: Learning universal image-text representations. In: European Conference on Computer Vision. Springer (2020)
9. Cheng, B., Schwing, A.G., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. arXiv preprint arXiv:2107.06278 (2021)
10. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016)
11. Das, A., Xian, Y., He, Y., Schiele, B., Akata, Z.: (sp)2net for generalized zero-label semantic segmentation. In: Pattern Recognition: 43rd DAGM German Conference, DAGM GCPR 2021. p. 235–249 (2021)
12. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
13. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
14. Everingham, M., Winn, J.: The pascal visual object classes challenge 2012 (voc2012) development kit. Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep **8**, 5 (2011)
15. Gao, T., Yao, X., Chen, D.: Simcse: Simple contrastive learning of sentence embeddings. arXiv preprint arXiv:2104.08821 (2021)
16. Ghiasi, G., Gu, X., Cui, Y., Lin, T.Y.: Open-vocabulary image segmentation. arXiv preprint arXiv:2112.12143 (2021)

17. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
18. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 580–587 (2014)
19. Gu, X., Lin, T.Y., Kuo, W., Cui, Y.: Zero-shot detection via vision and language knowledge distillation. arXiv preprint arXiv:2104.13921 (2021)
20. Gu, Z., Zhou, S., Niu, L., Zhao, Z., Zhang, L.: Context-aware feature generation for zero-shot semantic segmentation. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 1921–1929 (2020)
21. Gu, Z., Zhou, S., Niu, L., Zhao, Z., Zhang, L.: From pixel to patch: Synthesize context-aware features for zero-shot semantic segmentation. arXiv preprint arXiv:2009.12232 (2020)
22. Gupta, A., Dollar, P., Girshick, R.: Lvis: A dataset for large vocabulary instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5356–5364 (2019)
23. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
24. Hu, P., Sclaroff, S., Saenko, K.: Uncertainty-aware learning for zero-shot semantic segmentation. Advances in Neural Information Processing Systems **33** (2020)
25. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q.V., Sung, Y., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. arXiv preprint arXiv:2102.05918 (2021)
26. Kato, N., Yamasaki, T., Aizawa, K.: Zero-shot semantic segmentation via variational mapping. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. pp. 0–0 (2019)
27. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems. vol. 25. Curran Associates, Inc. (2012)
28. Lampert, C.H., Nickisch, H., Harmeling, S.: Attribute-based classification for zero-shot visual object categorization. IEEE transactions on pattern analysis and machine intelligence **36**(3), 453–465 (2013)
29. Li, A., Jabri, A., Joulin, A., Van Der Maaten, L.: Learning visual n-grams from web data. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4183–4192 (2017)
30. Li, B., Weinberger, K.Q., Belongie, S., Koltun, V., Ranftl, R.: Language-driven semantic segmentation. In: International Conference on Learning Representations (2022)
31. Li, G., Duan, N., Fang, Y., Gong, M., Jiang, D.: Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 11336–11344 (2020)
32. Li, P., Wei, Y., Yang, Y.: Consistent structural relation learning for zero-shot segmentation. Advances in Neural Information Processing Systems **33** (2020)
33. Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., et al.: Oscar: Object-semantics aligned pre-training for vision-language tasks. In: European Conference on Computer Vision. pp. 121–137. Springer (2020)
34. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. arXiv preprint arXiv:2107.13586 (2021)

35. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030 (2021)
36. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)
37. Lu, J., Batra, D., Parikh, D., Lee, S.: Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. arXiv preprint arXiv:1908.02265 (2019)
38. Lv, F., Liu, H., Wang, Y., Zhao, J., Yang, G.: Learning unbiased zero-shot semantic segmentation networks via transductive transfer. IEEE Signal Processing Letters **27**, 1640–1644 (2020)
39. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The role of context for object detection and semantic segmentation in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 891–898 (2014)
40. Pastore, G., Cermelli, F., Xian, Y., Mancini, M., Akata, Z., Caputo, B.: A closer look at self-training for zero-label semantic segmentation (2021)
41. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. arXiv preprint arXiv:2103.00020 (2021)
42. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
43. Song, J., Shen, C., Yang, Y., Liu, Y., Song, M.: Transductive unbiased embedding for zero-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1024–1033 (2018)
44. Su, W., Zhu, X., Cao, Y., Li, B., Lu, L., Wei, F., Dai, J.: Vl-bert: Pre-training of generic visual-linguistic representations. arXiv preprint arXiv:1908.08530 (2019)
45. Tian, G., Wang, S., Feng, J., Zhou, L., Mu, Y.: Cap2seg: Inferring semantic and spatial context from captions for zero-shot image segmentation. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 4125–4134 (2020)
46. Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. International journal of computer vision **104**(2), 154–171 (2013)
47. Xian, Y., Akata, Z., Sharma, G., Nguyen, Q., Hein, M., Schiele, B.: Latent embeddings for zero-shot classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 69–77 (2016)
48. Xian, Y., Choudhury, S., He, Y., Schiele, B., Akata, Z.: Semantic projection network for zero-and few-label semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8256–8265 (2019)
49. Yin, M., Yao, Z., Cao, Y., Li, X., Zhang, Z., Lin, S., Hu, H.: Disentangled non-local neural networks. In: European Conference on Computer Vision. pp. 191–207. Springer (2020)
50. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 633–641 (2017)
51. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. arXiv preprint arXiv:2109.01134 (2021)