A Simple Baseline for Zero-shot Semantic Segmentation with Pre-trained Vision-language Model

Mengde Xu\textsuperscript{1,3*}, Zheng Zhang\textsuperscript{1,3*}, Fangyun Wei\textsuperscript{3*}
Yutong Lin\textsuperscript{2,3}, Yue Cao\textsuperscript{3}, Han Hu\textsuperscript{3}, Xiang Bai\textsuperscript{1}

\textsuperscript{1}Huazhong University of Science and Technology
\textsuperscript{2}Xi’an Jiaotong University
\textsuperscript{3}Microsoft Research Asia

Abstract

Recently, zero-shot image classification by vision-language pre-training has demonstrated incredible achievements, that the model can classify arbitrary category without seeing additional annotated images of that category. However, it is still unclear how to make the zero-shot recognition working well on broader vision problems, such as object detection and semantic segmentation. In this paper, we target for zero-shot semantic segmentation, by building it on an off-the-shelf pre-trained vision-language model, i.e., CLIP. It is difficult because 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 on 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 zero-shot classification on the masked image crops which are generated in the first stage. Our experimental results show that this simple framework surpasses previous state-of-the-arts by a large margin: +29.5 \text{hIoU} on the Pascal VOC 2012 dataset, and +8.9 \text{hIoU} on the COCO Stuff dataset. With its simplicity and strong performance, we hope this framework to serve as a baseline to facilitate the 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 \textsuperscript{[16,24,29,36,45]} of deep learning, the semantic segmentation has also evolved tremendously under the supervised learning paradigm \textsuperscript{[3,9,37]}. However, unlike 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 segmenters’ capability in handling rich semantics. This issue can be alleviated by few-shot/one-shot/zero-shot algorithms \textsuperscript{[7,15,25,44]}, and this paper focuses on the most basic zero-shot settings which aim to segment novel categories that have no single segmentation annotation.

Existing works on zero-shot semantic segmentation \textsuperscript{[7,40,51,51]} are usually based on fully convolutional networks (FCN) \textsuperscript{[37]}, an architecture widely used for fully supervised semantic segmentation. In FCN, semantic segmentation is modeled 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. In zero-shot semantic segmentation, that classifier column of a novel category cannot be learnt from training data since there is no single pixel annotation of this category. Thus usually a generalizable metric between the category semantic and pixel features is learnt to seize such zero-shot capability \textsuperscript{[48,51]}. However, this pixel-level vision-category correspondence/metric is difficult to be well learnt given only tens or hundreds of categories involved by the semantic segmentation tasks.

In this work, we propose to leverage a recent advance of image-level vision-language learning model, i.e., CLIP \textsuperscript{[41]}, to address this issue, which has learnt a strong vision-category alignment model using rich image-caption data. A main difficulty in leveraging this model lies in that
the CLIP model is learnt at image-level, which is different in granularity than that of the pixel-level semantic segmentation task. One way to bridge this granularity difference is to reformulate the image-level CLIP model to a pixel-level one such that an alignment between pixel-level vision features and the text feature can be converted from the image-level alignment. This pixel-level vision-category alignment model is then leveraged for the FCN based zero-shot semantic segmentation. However, the pixel-level alignment performs unsatisfactorily due to that the model is learnt in image-level.

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 [11], which first extracts a set of class-agnostic mask proposals and then classifies each mask proposal into a specific 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-adaptable to handle novel categories: firstly, the class-agnostic mask proposal generation trained using seen categories is observed well generalizable to novel categories; 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 zero-shot categorization. In addition, we employ a prompt-learning approach [35] to further improve the zero-shot classification accuracy given a pre-trained CLIP model.

We show that the proposed approach, though simple and straightforward, can surpass previous state-of-the-arts [7, 21, 40, 51] by a large margin. On Pascal VOC 2012 [17], 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 [8], 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 [10,31,33,38,47] was performs 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 [27, 41] 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 image classification and image-text retrieval.

This work introduces the CLIP model as a strong vision-category correspondent to improve the zero-shot 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 [37] and its variants [3, 9, 52], 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 [11] explored to model the semantic segmentation as two sub-tasks: segment generation and segment classification and have shown competitive performance compared to FCN based approaches.

Zero-shot Learning Zero-shot learning for image classification are widely studied in recent years. Most explorations [1, 50] works on learning a joint embedding space between the images and the name/description of the category. Some other works explore taking the advantages of mid-level semantic representation [30], using the synthetic data [6], or leveraging additional unlabeled data from seen classes [46].

Another important application of zero-shot learning is object detection [5, 13, 34, 42]. Most advanced methods of zero-shot object detection rely on generalized object proposals, such as RPN [43] or EdgeBox [55], and converting the detecting problem into a classification problem. Most recently, [20] introduce the CLIP into the zero-shot object detection and show it could significantly improve the long-tile object detection [23].

Zero-shot Semantic Segmentation Some pioneer works to study the zero-shot learning for semantic segmentation. ZS3Net [7] uses generative models to synthesize pixel-level features by word embeddings of unseen classes. CSRL [32] further incorporating the structural relation in feature synthesis. CaGNet [21, 22] introduce a contextual module for better feature generation. Different from [7, 21, 32], SPNet [51] 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 [28], variational mapping is used to learn semantic features. In [26], the uncertainty-aware losses are proposed to eliminate noisy samples. Other works explored other directions or aspects of zero-shot semantic segmentation. In [40], the self-training for zero-shot semantic segmentation are carefully studied. In [39, 46], the transductive learn-
ing setting are explored. In [48], 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.

3. Preliminary

We first introduce the setting of zero-shot semantic segmentation and revisit CLIP in this section as preliminary.

3.1. Zero-shot Semantic Segmentation

Zero-shot semantic segmentation aims to predict masks for unseen classes $C_{\text{unseen}}$ by learning from some labeled data of seen classes $C_{\text{seen}}$ and the seen classes and unseen classes are disjoint, i.e., $C_{\text{unseen}} \cap C_{\text{seen}} = \emptyset$. Usually, $C_{\text{seen}}$ and $C_{\text{unseen}}$ are often represented with semantic words like dog, cat, apple, and sometimes the description of the classes are also provided.

There are various settings for zero-shot semantic segmentation due to the differences in evaluation protocols or data split manner. Among them, the most widely used setting [7, 40, 51] is Generalized Zero-shot Semantic Segmentation. In this setting, a training set $X_{\text{train}} = \{(I_k, M_k)\}$ with input images $I_k$ and the ground-truth semantic segmentation annotations $M_k$ are provided for model training, and a testing set $X_{\text{test}}$ are used for evaluation. In particular, the annotations $\{M_k\}$ of the training set $X_{\text{train}}$ contains only the seen classes, while both seen classes and unseen classes need to be predicted in testing set $X_{\text{test}}$. In this work, we target this setting for evaluating our method.

3.2. Revisiting CLIP

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

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 image classification. Specifically, the class names are injected into the pre-defined prompt template and fed into CLIP text encoder generate the class embedding, 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 embedding are used as the classifier and the similarity with image embedding is computed to perform classification.

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

4. Two-Stage Zero-shot Semantic Segmentation

Figure 1 shows an overview of our two-stage zero-shot semantic segmentation framework, and we will describe each component of this framework in the following.

4.1. Mask Proposal Generation

We first introduce the mask proposal generation. In our work, we tried three different methods to generate the mask proposals $\{M_k\}$:

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 [49]. This method can also generate hierarchical segments. Since this method can effectively localize objects, it is widely used in object detection system [18, 19].
**MaskFormer** [11]. 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 suggest that the predicted segments by MaskFormer can be used as the mask proposals, and we empirically demonstrate (see Table. 5) 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

#### 4.2.1 Two Strategies for Using CLIP

There are two strategies to perform the region classification by utilizing the CLIP pre-trained model:

- The first strategy is to directly apply the CLIP vision encoder on each mask proposals for classification. Specifically, given an image $I$ and a mask proposal $M^p$, we first binarized the mask proposal by setting a threshold of 0.5, and then applying the binarized $M^p$ to image $I$, erase the unused background and only crop foreground area. The masked image crop is resized to $224^2$ and sent to CLIP for classification. However, since there is no extra training process in this approach, the training data of seen classes cannot be utilized, and resulting in inferior performance on seen classes in the inference (see Table. 6).

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

The two strategies complement each other (see Table. 6), therefore we ensemble the results of these two strategies by default.

#### 4.2.2 Text Prompting

The pre-training of the CLIP model is not designed for zero-shot semantic segmentation, and how to construct feasible text prompts need to be explored.

**Hand-crafted Prompting** A simple approach is to reuse the hand-crafted prompts provided by CLIP which is originally designed for image classification on ImageNet-1K [14]. 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 an adverse effect. So we evaluate each of these prompts on training data to select one most helpful prompt for zero-shot semantic segmentation.

**Learning-based Prompt** The learned-based prompt [35, 54] recently show great potential for adapting the large-scale pre-training language/vision-language models on specific downstream tasks. We also explored this technique in our approach. Specifically, an input sentence of text encoder in CLIP model is a sequence of token, and this token sequence contains two token type: $[p]$ indicates the prompt token and $[cat]$ indicates the category token. A generalized prompted text can be formulated as $[p_0][p_1][cat_0]...[cat_n][p_m]$, where $n$ is the number of category token and $m$ is the number of prompt token. In learning-based prompt, the prompt tokens $[p_0][p_1]...[p_m]$ are a set of learnable parameters that can be trained on the data of seen classes and can generalize to unseen classes.

### 4.3. Mask Prediction Assembly

Because 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 point $q$, its predicted probability of being $i$-th category are defined as:

$$C_i(q) = \frac{M^p_k(q)C^p_k(i)}{\sum_k M^p_k(q)},$$  \hspace{1cm} (1)

where $M^p_k(q)$ denotes the predicted probability of point $q$ in $k$-th mask proposal $M^p_k$, and $C^p_k(i)$ is the predicted probability of mask proposals $M^p_k$ belonging to $i$-th category. Note that the sum of $C_i(q)$ over all categories is not guaranteed to be 1, and point $q$ are 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 fully convolution net-
work (FCN), which is widely used by previous works. As a dominant method in supervising semantic segmentation, FCN formulated the semantic segmentation as a pixel-wise classification problem. Specifically, a feature map with a spatial resolution are generated by FCN for a given image, and a set of learned classifiers is applied to each pixel of the feature map, producing the segmentation results.

Similar to our proposed two-stage framework, there are also two strategies to use the CLIP model in 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 [9] for performing multi-scale inference. We empirically found that it can improve performance (see Table. 3).

- The training data of the seen classes cannot be utilized in the first approach. Instead, we can retrain an FCN-based vision encoder on seen classes via the similar method introduced in Sec. 4.2.1. 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

6.1.1 Dataset

We conduct extensive experiments on four challenging datasets to evaluate our method: COCO Stuff [8], Pascal VOC 2012 [17], Cityscapes [12], and ADE20K [53].

COCO Stuff This is a large-scale dataset used in MS COCO Challenge, containing 117K training images, 5k validation images. Following the setting in [51], a total of 171 annotated classes are divide into 156 seen classes and 15 unseen classes.

Pascal VOC 2012 This dataset contains 11185 training images and 1449 validation images. Following [51], a total of 20 classes are divided into 15 seen classes and 5 unseen classes. The provided augmented annotations are used.

ADE20K This dataset contains 20K training images, 2K validation images, and 3K testing images. There are a total of 150 classes, including both indoors and outdoors.

Cityscapes This is a scene parsing dataset collected on urban streets, containing 5000 finely annotated images and 20000 coarsely annotated images. The finely annotated set is divided into 2975/500/1525 splits for training, validation, and testing, respectively. There are 30 classes annotations in total, while we use only 19 classes for evaluation according to the common practices [12].

6.1.2 Evaluation Protocol

Following previous works [40, 51], we use the mean of class-wise intersection (mIoU) over union on unseen classes and harmonic mean IoU (hIoU) among the seen classes and unseen classes as our major metric. The harmonic mean IoU is defined as:

$$hIoU = \frac{2 \times mIoU_{seen} \times mIoU_{unseen}}{mIoU_{seen} + mIoU_{unseen}}.$$  (2)

In addition, we also report the pixel-wise classification accuracy (pAcc) as a reference.

6.2. Implementation Details

We conduct all the experiments on 8 × Nvidia V100 GPUs. We train a MaskFormer model on COCO Stuff dataset with ResNet-101 as 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. If not specified, the MaskFormer model only trained on seen classes and ignore the unseen classes, and we use 100 mask proposals for both training and testing. For all other settings and hyper-parameters, we keep the official setting of [11] without changes.

For the CLIP model, the ViT-B/16 backbone is used by default if not specified. In text prompt tuning, the prompts are randomly initialized, and a Stochastic Gradient Descent (SGD) optimizer is used to train the 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 with 100, 50 training epochs for COCO Stuff and Pascal VOC, respectively.

For Pascal VOC 2012 dataset, we use a batch size of 16 and a total training iterations of 20K, and keep all other setting as the same as COCO Stuff dataset.
### Table 1. Comparison with state-of-the-art methods on the validation set of Pascal VOC dataset.

| Method          | hIoU  | pAcc | mIoU  |
|-----------------|-------|------|-------|
| SPNet [51]      | 21.8  | -    | 73.3  |
| ZS3 [7]         | 28.7  | -    | 77.3  |
| CaGNet [21]     | 39.7  | -    | 78.4  |
| Ours            | 77.5  | -    | **83.5** |
| SPNet+ST [51]   | 38.8  | -    | 77.80 |
| ZS3 [7]         | 33.3  | -    | 78.1  |
| CaGNet+ST [21]  | 43.7  | -    | 78.6  |
| STRICT [40]     | 49.8  | -    | 82.7  |
| Ours+ST         | **79.3** | 88.7 | **78.1** |

Table 2. Comparison with state-of-the-art methods on the validation set of COCO Stuff dataset.

| Method          | hIoU  | pAcc | mIoU  |
|-----------------|-------|------|-------|
| FCN [37]        | 11.7  | -    | 20.5  |
| FCN + SW [37]   | 20.9  | -    | 34.7  |
| Ours            | 37.7  | 60.3 | **39.6** |

6.3. Comparison with State-of-the-Arts

We first 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 reported the performance w/ or w/o self-training in the following.

**Pascal VOC 2012** The results are shown in Table 1. Without using the self-training, our method achieved 77.5 hIoU and 72.5 mIoU-unseen, outperforming the previous best method CaGNet by a huge margin of +37.7 on hIoU and +46.8 on mIoU-unseen. Further employing the self-training, our method achieved 79.3 on hIoU and 72.5 on mIoU-unseen, outperforming the previous best method STRICT by +29.5 on hIoU and +42.5 on mIoU-unseen.

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

6.4. 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 MaskFormer model of ResNet-101 and CLIP of ViT-B/16.

6.4.1 Comparison with Fully Convolutional Network

We first compared the FCN approach with proposed two-stage approach. For a fair comparison, we use ResNet-101 in FCN and ViT-B/16 in CLIP. Table 3 shows the results, the plain FCN only achieved 11.7 hIoU and 10.4 mIoU-unseen. Further employing the sliding window on FCN, performance is significantly improved by +9.2 on hIoU and +5.6 on mIoU-unseen, indicating that the image size inconsistency between pre-training and testing of CLIP model does affect the performance. However, even equipped with the sliding window, the performance of FCN approach still worse than our two-stage approach by -16.8 on hIoU and -20.3 on mIoU-unseen, indicating that our two-stage framework is more suitable for the CLIP model.

---

Table 3. Zero shot semantic segmentation performance of a FCN-style method on COCO Stuff dataset. SW: Sliding Window Testing, each image is split into several $224 \times 224$ patches.

| Method          | hIoU  | pACC | mIoU  |
|-----------------|-------|------|-------|
| FCN [37]        | 11.7  | 54.9 | 10.4  |
| FCN + SW [37]   | 20.9  | 50.8 | 16.0  |
| Ours            | 37.7  | 60.3 | **36.3** |

Figure 2. Qualitative result of our method on COCO Stuff dataset. Only results of unseen classes are visualized. Predictions misclassified to seen classes are labeled with *seen* color.
### Table 4. Ablation study of different proposal generation methods on COCO Stuff dataset.

| Method                  | hIoU  | pAcc  | mIoU-unseen |
|-------------------------|-------|-------|-------------|
| GPB-UCM [2]             | 10.9  | 9.5   | 11.6        |
| Selective Search [49]   | 11.0  | 23.5  | 13.3        |
| MaskFormer [11]         | 28.2  | 48.4  | 29.7        |

### Table 5. Evaluating 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  |

6.4.2 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 on trained classifier on seen classes, we perform the comparison by directly performing CLIP model on mask proposals. The results are shown in Table 4, and the MaskFormer achieved better performance than the Selective Search and GPB-UCM. Note that even the other two methods are worse than MaskFormer, they still achieved comparable performance compared with state-of-the-arts on mIoU-unseen.

6.4.3 Generalization of Mask Proposal Generator

Although using MaskFormer to generate the mask proposals achieves excellent performance in training on seen classes and testing on unseen classes under the same dataset, it is still unknown whether Maskformer can produce good performance on cross-dataset setting, i.e., training on one dataset and testing on another dataset. Therefore, we evaluate the generalization ability of using MaskFormer to generate mask proposals between the COCO Stuff dataset and ADE20K dataset.

In this experiment, we want to evaluate only the quality of 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 5, and we directly report the mIoU in this experiment because the seen class cannot be defined between different dataset. 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 an acceptable performance degradation, and vice versa. That demonstrates the generalization ability to use MaskFormer as the proposal generator.

6.4.4 Different Strategies for Using CLIP

We study the two different strategies for using CLIP discussed in Sec. 4.2.1: Retrained vision encoder or directly using CLIP vision encoder without tuning. The results are shown in Table 6. 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 has significantly improved, indicating the two strategies are complementary.

6.4.5 Text Prompting

We compared two prompt tuning methods described in Sec. 4.2.2. The results are shown in Table 7. The learned prompt outperforms the manually searched prompts by +9.9 hIoU, clearly showing the power of learned prompts. In addition, although the learned prompts are only trained on seen classes, we noticed that it achieved similar improvement on seen classes and unseen classes(+9.6 mIoU-seen and +10.2 mIoU-unseen), indicating the learned prompt has a strong generalization ability to the unseen class.

We further study how prompt length and training data size affects the performance of learned prompt by training on seen classes and testing on unseen classes. The results are shown in Table 8, and the best performance is achieved when we use 32 samples for each category, in either prompt length of 16 or 32, and more training samples will degrade the performance, we speculate that more samples may lead
| Text Prompt | hIoU | pAcc | mIoU  |
|-------------|------|------|-------|
| Manual      | 18.3 | 28.6 | 17.3  |
| Learned     | 28.2 | 48.4 | 26.8  |

Table 7. Manually designed text prompt v.s. Learned text prompt.

| Prompt Len. | Sample Num. | Unseen Cls. Acc |
|-------------|--------------|-----------------|
| 16          | 16           | 28.9            |
|             | 32           | 30.0            |
|             | 64           | 29.5            |
|             | all          | 25.5            |
| 32          | 16           | 32.1            |
|             | 32           | 32.8            |
|             | 64           | 31.0            |
|             | all          | 27.6            |

Table 8. We study the effect of prompt length and sample number of each category for prompt learning.

| Backbone | hIoU | pAcc | mIoU-unseen |
|----------|------|------|-------------|
| ResNet50 | 15.2 | 22.2 | 16.0        |
| ResNet101| 13.8 | 19.7 | 12.5        |
| ViT-B/32 | 15.3 | 25.2 | 15.7        |
| ViT-B/16 | 18.3 | 28.6 | 19.5        |

Table 9. Evaluate the performance of different CLIP variants in our framework on COCO Stuff dataset with manual prompt setting.

over-fitting issue, which is also reported by other prompt learning attempts [54].

6.4.6 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 classifier. We report the number without using the learned text prompt since the high experimental overhead. The results are shown in Table 9. We find all models perform well and ViT-B/16 achieved the best performance.

6.4.7 Transferability to Other Datasets

We evaluate the transferability of our approach across dataset. We train our method only on the seen classes of COCO Stuff and directly test it on other datasets without finetuning. The results are shown in Table 10. The model trained on COCO Stuff can well transferred to Pascal VOC with strong performance, and reach a reasonable performance on Cityscapes and ADE20K.

6.4.8 Comparison with Supervised Baseline

We also compare our method with the supervised baseline on COCO Stuff. The supervised models are MaskFormer with ResNet-101 backbone trained on all classes, including seen and unseen classes. We report the result in Table 11. 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 result is reported in Table 12. We found 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 that work, we propose a simple and effective two-stage framework for zero-shot semantic segmentation with the advanced pre-trained vision-language model, i.e., CLIP.
We reformulate and break down the zero-shot semantic segmentation into two steps: training a mask proposal generator to generate a set of binary masks and leveraging the pre-trained CLIP to classify each mask proposal. We conduct extensive experiments to verify the advantages and limitations of applying the pre-trained vision-language model to zero-shot semantic segmentation. Notably, the proposed framework outperforms previous state-of-the-arts on Pascal VOC 2012 and COCO Stuff by large margins. Our work reveals the potential for using pre-trained vision-language models on zero-shot semantic segmentation and provides a strong baseline for this community to facilitate the future research.

References

[1] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for image classification. IEEE transactions on pattern analysis and machine intelligence, 38(7):1425–1438, 2015.

[2] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. IEEE transactions on pattern analysis and machine intelligence, 33(5):898–916, 2010.

[3] 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.

[4] Donghyeon Baek, Youngmin Oh, and Bumsub Ham. Exploiting a joint embedding space for generalized zero-shot semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 9536–9545, 2021.

[5] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In Proceedings of the European Conference on Computer Vision (ECCV), pages 384–400, 2018.

[6] Maxime Bucher, Stéphane Herbin, and Frédéric Jurie. Generating visual representations for zero-shot classification. In Proceedings of the IEEE International Conference on Computer Vision Workshops, pages 2666–2673, 2017.

[7] Maxime Bucher, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Zero-shot semantic segmentation. Advances in Neural Information Processing Systems, 32:468–479, 2019.

[8] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1209–1218, 2018.

[9] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. 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.

[10] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Learning universal image-text representations. In European Conference on Computer Vision. Springer, 2020.

[11] Bowen Cheng, Alexander G Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. arXiv preprint arXiv:2107.06278, 2021.

[12] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3213–3223, 2016.

[13] Berkan Demirel, Ramazan Gokberk Cinbis, and Nazli Ikizler-Cinbis. Zero-shot object detection by hybrid region embedding. arXiv preprint arXiv:1805.06157, 2018.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.

[15] Nanqing Dong and Eric P Xing. Few-shot semantic segmentation with prototype learning. In BMVC, volume 3, 2018.

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.

[17] Mark Everingham and John Winn. The pascal visual object classes challenge 2012 (voc2012) development kit. Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep. 8:5, 2011.

[18] Ross Girshick. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 1440–1448, 2015.

[19] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 580–587, 2014.

[20] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Zero-shot detection via vision and language knowledge distillation. arXiv preprint arXiv:2104.13921, 2021.

[21] Zhangxuan Gu, Siyuan Zhou, Li Niu, Zihan Zhao, and Liqing Zhang. Context-aware feature generation for zero-shot semantic segmentation. In Proceedings of the 28th ACM International Conference on Multimedia, pages 1921–1929, 2020.

[22] Zhangxuan Gu, Siyuan Zhou, Li Niu, Zihan Zhao, and Liqing Zhang. From pixel to patch: Synthesize context-aware features for zero-shot semantic segmentation. arXiv preprint arXiv:2009.12232, 2020.

[23] Agrim Gupta, Piotr Dollár, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5356–5364, 2019.
[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[25] Seunghoon Hong, Hyeonwoo Noh, and Bohyung Han. Decoupled deep neural network for semi-supervised semantic segmentation. arXiv preprint arXiv:1506.04924, 2015.

[26] Ping Hu, Stan Sclaroff, and Kate Saenko. Uncertainty-aware learning for zero-shot semantic segmentation. Advances in Neural Information Processing Systems, 33, 2020.

[27] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. arXiv preprint arXiv:2102.05918, 2021.

[28] Naoki Kato, Toshihiko Yamasaki, and Kiyoharu Aizawa. Zero-shot semantic segmentation via variational mapping. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, pages 0–0, 2019.

[29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems, volume 25. Curran Associates, Inc., 2012.

[30] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. IEEE transactions on pattern analysis and machine intelligence, 36(3):453–465, 2013.

[31] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In Proceedings of the AAAI Conference on Artificial Intelligence, 2021.

[32] Peike Li, Yunchao Wei, and Yi Yang. Consistent structural relation learning for zero-shot segmentation. Advances in Neural Information Processing Systems, 33, 2020.

[33] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furui Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In European Conference on Computer Vision, pages 121–137. Springer, 2020.

[34] Zhihui Li, Lina Yao, Xiaojin Zhang, Xianzhi Wang, Salil Kanhere, and Huaxiang Zhang. Zero-shot object detection with textual descriptions. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 8690–8697, 2019.

[35] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbo Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. arXiv preprint arXiv:2107.13586, 2021.

[36] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030, 2021.

[37] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3431–3440, 2015.

[38] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefano. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. arXiv preprint arXiv:1908.02265, 2019.

[39] Fengmiao Lv, Haiyang Liu, Yichen Wang, Jiayi Zhao, and Guowu Yang. Learning unbiased zero-shot semantic segmentation networks via transductive transfer. IEEE Signal Processing Letters, 27:1640–1644, 2020.

[40] Giuseppe Pastore, Fabio Cermelli, Yongqin Xian, Massimiliano Mancini, Zeynep Akata, and Barbara Caputo. A closer look at self-training for zero-label semantic segmentation, 2021.

[41] 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. arXiv preprint arXiv:2103.00020, 2021.

[42] Shafin Rahman, Salman Khan, and Fatih Porikli. Zero-shot object detection: Learning to simultaneously recognize and localize novel concepts. In Asian Conference on Computer Vision, pages 547–563. Springer, 2018.

[43] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28:91–99, 2015.

[44] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. arXiv preprint arXiv:1709.03410, 2017.

[45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[46] Jie Song, Chengchao Shen, Yezhou Yang, Yang Liu, and Mingli Song. Transductive unbiased embedding for zero-shot learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1024–1033, 2018.

[47] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furui Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visiolinguistic representations. arXiv preprint arXiv:1908.08530, 2019.

[48] Guiyu Tian, Shuai Wang, Jie Feng, Li Zhou, and Yadong Mu. Cap2seg: Inferring semantic and spatial context from captions for zero-shot image segmentation. In Proceedings of the 28th ACM International Conference on Multimedia, pages 4125–4134, 2020.

[49] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. International journal of computer vision, 104(2):154–171, 2013.

[50] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 69–77, 2016.
[51] 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. 1, 2, 3, 5, 6

[52] Minghao Yin, Zhuliang Yao, Yue Cao, Xiu Li, Zheng Zhang, Stephen Lin, and Han Hu. Disentangled non-local neural networks. In *European Conference on Computer Vision*, pages 191–207. Springer, 2020. 2

[53] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 5

[54] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *arXiv preprint arXiv:2109.01134*, 2021. 4, 8

[55] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European conference on computer vision*, pages 391–405. Springer, 2014. 2