# Decoupling Zero-Shot Semantic Segmentation

Jian Ding[1,2], Nan Xue[1], Gui-Song Xia[1]*, Dengxin Dai[2]
[1]CAPTAIN, Wuhan University, China [2]MPI for Informatics, Germany
{jian.ding, xuenan, guisong.xia}@whu.edu.cn, ddai@mpi-inf.mpg.de

## Abstract

*Zero-shot semantic segmentation (ZS3) aims to segment the novel categories that have not been seen in the training. Existing works formulate ZS3 as a pixel-level zero-shot classification problem, and transfer semantic knowledge from seen classes to unseen ones with the help of language models pre-trained only with texts. While simple, the pixel-level ZS3 formulation shows the limited capability to integrate vision-language models that are often pre-trained with image-text pairs and currently demonstrate great potential for vision tasks. Inspired by the observation that humans often perform segment-level semantic labeling, we propose to decouple the ZS3 into two sub-tasks: 1) a class-agnostic grouping task to group the pixels into segments. 2) a zero-shot classification task on segments. The former task does not involve category information and can be directly transferred to group pixels for unseen classes. The latter task performs at segment-level and provides a natural way to leverage large-scale vision-language models pre-trained with image-text pairs* (e.g. *CLIP) for ZS3. Based on the decoupling formulation, we propose a simple and effective zero-shot semantic segmentation model, called ZegFormer, which outperforms the previous methods on ZS3 standard benchmarks by large margins,* e.g., ***22 points** on the PASCAL VOC and **3 points** on the COCO-Stuff in terms of mIoU for unseen classes. Code will be released at* https://github.com/dingjiansw101/ZegFormer.

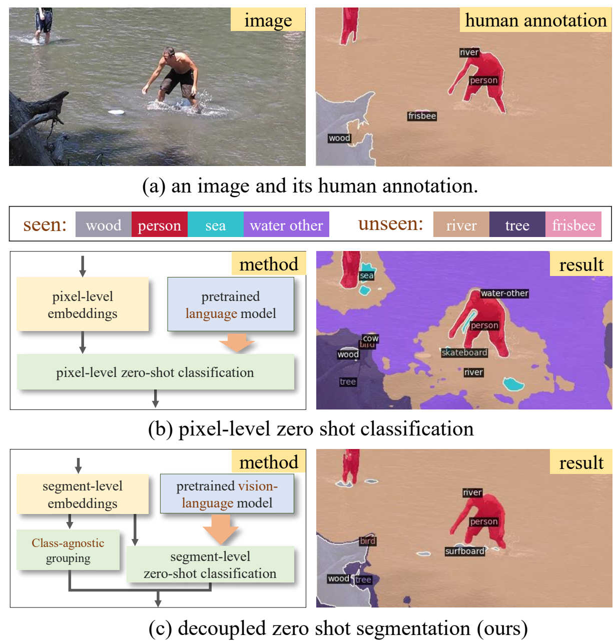


Figure 1. ZS3 aims to train a model merely on *seen classes* and generalize it to classes that have not been seen in the training (*unseen classes*). Existing methods formulate it as a pixel-level zero-shot classification problem **(b)**, and use semantic features from a *language* model to transfer the knowledge from seen classes to unseen ones. In contrast, as in **(c)**, we decouple ZS3 into two sub-tasks: 1) A *class-agnostic* grouping and 2) A *segment-level* zero-shot classification, which enables us to take full advantage of the pre-trained *vision-language* model.

## 1. Introduction

Semantic segmentation targets to group an image into segments with semantic categories. Although remarkable progress has been made [10, 11, 36, 56, 57, 62], current semantic segmentation models are mostly trained in a supervised manner with a fixed set of predetermined semantic categories, and often require hundreds of samples for each class. In contrast, humans can distinguish at least 30,000 basic categories [6, 18], and recognize novel categories merely from some *high-level descriptions*. How to achieve human-level ability to recognize stuff and things in images is one of the ultimate goals in computer vision.

Recent investigations on zero-shot semantic segmentation (ZS3) [7, 54] have actually moved towards that ultimate goal. Following the fully supervised semantic segmentation models [10, 11, 36] and zero-shot classification models [1, 24, 30, 47, 60], these works formulate zero-shot semantic segmentation as a *pixel-level zero-shot classification problem*. Although these studies have reported promising results, two main issues still need to be addressed : (1) They usually transfer knowledge from seen to unseen classes by *language* models [7, 27, 54] pre-trained only by texts, which

*Corresponding author

limit their performance on vision tasks. Although large-scale pre-trained *vision-language* models (*e.g*. CLIP [46] and ALIGN [26]) have demonstrated potentials on *image-level* vision tasks, how to efficiently integrate them into the *pixel-level* ZS3 problem is still unknown. (2) They usually build correlations between *pixel-level* visual features and semantic features for knowledge transfer, which is *not natural* since we humans often use *words or texts* to describe *objects/segments* instead of *pixels* in images. As illustrated in Fig. 1, it is unsurprising to observe that the *pixel-level* classification has poor accuracy on *unseen classes*, which in turn degrades the final segmentation quality. This phenomenon is particularly obvious when the number of unseen categories is large (see Fig. 6).

An intuitive observation is that, given an image for semantic segmentation, we humans can first *group pixels into segments* and then perform a *segment-level semantic labeling* process. For example, a child can easily group the pixels of an object, even though he/she does not know the name of the object. Therefore, we argue that *a human-like zero-shot semantic segmentation procedure should be decoupled into two sub-tasks*:

- A *class-agnostic grouping* to group pixels into segments. This task is actually a classical image partition/grouping problem [44,50,52], and can be renewed via deep learning based methods [12,32,59].
- A *segment-level zero-shot classification* to assign semantic labels either seen or unseen to segments.

As the the grouping task does not involve the semantic categories, a grouping model learned from seen classes can be easily transferred to unseen classes. The segment-level zero-shot classification is robust on the unseen classes and provides a flexible way to integrate the pre-trained large-scale vision-language models [46] to the ZS3 problem.

To instantiate the decoupling idea, we present a simple yet efficient *zero-shot semantic segmentation model with transformer*, named ZegFormer, which uses a transformer decoder to output *segment-level* embeddings, as shown in Fig. 2. It is then followed by a *mask projection* for *class-agnostic grouping* (CAG) and a *semantic projection* for *segment-level zero-shot classification* (s-ZSC). The *mask projection* maps each *segment-level* embedding to a mask embedding, which can be used to obtain a binary mask prediction via a dot product with a high-resolution feature map. The *semantic projection* establishes the correspondences between *segment-level* embedding and semantic features of a pre-trained *text encoder* for s-ZSC.

While the steps mentioned above can form a standalone approach for ZS3, the model trained on a small dataset is struggling to have strong generalization. Thanks to the decoupling formulation, it is also flexible to use an *image encoder* of a vision-language model to generate image embeddings for zero-shot segment classification. As we empirically find that the segment classification scores with image embeddings and s-ZSC are complementary. We fuse them to achieve the final classification scores for segments.

The proposed ZegFormer model has been extensively evaluated with experiments and demonstrated superiority on various commonly-used benchmarks for ZS3. It outperforms the state-of-the-art methods by **22 points** in terms of mIoU for unseen classes on the PASCAL VOC [15], and **3 points** on the COCO-Stuff [8]. Based on the challenging ADE20k-Full dataset [64], we also create a new ZS3 benchmark with **275 unseen classes**, the number of unseen classes in which are much larger than those in PASCAL-VOC (5 unseen classes) and COCO-Stuff (15 unseen classes). On the ADE20k-Full ZS3 benchmark, our performance is comparable to MaskFormer [12], a fully supervised semantic segmentation model.

Our contributions in this paper are three-fold:

- We propose a new formulation for the task of ZS3, by decoupling it into two sub-tasks, *a class-agnostic grouping* and *a segment-level zero-shot classification*, which provides a more natural and flexible way to integrate the pre-trained large-scale vision-language models into ZS3.
- With the new formulation, we present a simple and effective ZegFormer model for ZS3, which uses a transformer decoder to generate *segment-level* embeddings for grouping and zero-shot classification. To the best of our knowledge, the proposed ZegFormer is the first model taking full advantage of the pre-trained large-scale vision-language model (*e.g*. CLIP [46]) for ZS3.
- We achieved state-of-the-art results on standard benchmarks for ZS3. The ablation and visualization analyses show that the decoupling formulation is superior to pixel-level zero-shot classification by a large margin.

## 2. Related Works

**Zero-Shot Image Classification** aims to classify images of *unseen categories* that have not been seen during training. The key idea in zero-shot learning is to transfer kowledge from *seen classes* to *unseen classes* via semantic representation, such as the semantic attributes [2,24,28,30], concept ontology [16,37,39,47,48] and semantic word vectors [17, 41, 60]. Recently, there are some works that use large-scale *vision-language pretraining* [26, 46] via contrastive loss. For example, by training a vision-language model on 400 million image and text pairs collected by Google Engine *without any human annotations*, CLIP [46] has achieved impressive performances on more than 30 vision datasets even compared to the supervised models. It has also shown the potential for zero-shot object detection [19]. However, there is a large gap between the *pixel level features* used in previous ZS3 models and *image-level features* used in CLIP. To bridge this gap, we build the correspondence between *segment-level* visual features and *image-level* vision-language features.

**Zero-Shot Segmentation** is a relatively new research topic [7, 25, 42, 54, 61], which aims to segment the classes that have not been seen during training. There have been two streams of work: discriminative methods [5,42,54] and generative methods [7, 20, 49]. SPNet [54] and ZS3Net [7] are considered the representative examples. In detail, SPNet [54] maps each pixel to a semantic word embedding space and projects each pixel feature into class probability via a fixed semantic word embedding [27,38] projection matrix. ZS3Net [7] first train a generative model to generate pixel-wise features of unseen classes by word embeddings. With the synthetic features, the model can be trained in a supervised manner. Both of these two works formulate ZS3 as a *pixel-level zero-shot classification* problem. However, this formulation is not robust for ZS3, since the text embeddings are usually used to describe objects/segments instead of pixels. The later works [20, 21, 29, 31, 49, 51] all follow this formulation to address different issues in ZS3. In a weaker assumption that the unlabelled pixels from unseen classes are available in the training images, self-training [7,42] is widely used. Although promising performances are reported, self-training often needs to retrain a model whenever a new class appears. Different from the pixel-level zero-shot classification, we propose a new formulation for ZS3, by decoupling ZS3 into a class-agnostic learning problem on pixels and a segment-level zero-shot learning problem. Then we implement a ZegFormer for ZS3, which does not have a complicated training scheme and is flexible to transfer to new classes without retraining. A recent work [63] also uses region-level classification for *bounding boxes*. But it focuses on *instance segmentation* instead of *semantic segmentation*. Besides, it still predicts class-***aware*** masks. We are the first to use the region-level classification for *zero-shot semantic segmentation*.

**Class-Agnostic Segmentation** is a long-standing problem and has been extensively studied in computer vision [3,4,14,44,50,52,65]. There are evidences [22,23,43] that *class-agnostic* segmentation model learned from seen classes can be well transferred to *unseen classes* in the task of *instance segmentation*, under a partially supervised training paradigm. Recently, a class-agnostic segmentation task [45] called *entity segmentation (ES)* is proposed, which can predict segments for both thing and stuff classes. However, ES is an *instance-aware task*, which is different from *semantic segmentation*. In addition, entity segmentation [45] does not predict the detailed class names of unseen classes. Our work is inspired by the abovementioned findings, but we focus on the *semantic segmentation* of novel classes and also predict the detailed class names of unseen classes. With the formulation, our proposed method is simpler, flexible, and robust.

# 3. Methodology

## 3.1. Decoupling Formulation of ZS3

Given an image $I$ on the domain $\Omega = \{0, \dots, H-1\} \times \{0, \dots, W-1\}$, the semantic segmentation of $I$ can be defined as a process to find a pair of mappings $(\mathcal{R}, \mathcal{L})$ for $I$, where $\mathcal{R}$ groups the image domain into $N$ piece-wise "homogeneous" segments $\{R_i\}_{i=1}^N$, such that $\cup_{i=1}^N R_i = \Omega$ and $R_i \cap R_j = \emptyset$, if $i \neq j$, and $\mathcal{L}$ associates every segment $R_i \in \Omega$ with a semantic label $c \in C$, where $C$ is a predefined set of categories.

A fully-supervised semantic segmentation suggests that, one can learn such a pair of mappings $(\mathcal{R}, \mathcal{L})$ for $I$ from a large-scale semantic annotated dataset, *i.e*., $\mathbf{D} = \{I_k, \mathcal{R}_k, \mathcal{L}_k\}_{k=1}^K$. This type of methods are often with an assumption that the category set $C$ are closed, *i.e*., the categories appearing in testing images are well contained by $C$, which, however are usually violated in real-application scenarios.

Actually, if we denote $S$ as the category set of the annotated dataset $\mathbf{D}$, *i.e*., the seen classes, and $E$ as those appearing in the testing process, we have three types of settings for semantic segmentation,

- fully-supervised semantic segmentation: $E \subseteq S$,
- zero-shot semantic segmentation (ZS3): $S \cap E = \emptyset$,
- generalized ZS3 (GZS3) problem: $S \subset E$.

In this paper, we mainly address the problem of GZS3, and denote $U = E - E \cap S$ as the set of unseen classes.

**Relations to Pixel-Level Zero-Shot Classification.** Previous works [7, 20, 54] formulate ZS3 as a *pixel-level zero-shot classification* problem, where a model learned from pixel-level semantic labels of $S$ needs to be generalized to pixels of $U$. It can be considered as *a special case* of our decoupled formulation, where each pixel represents a segment $R_i$. Since the learning of $\mathcal{R}$ does not involve the semantic categories, our formulation separates a *class-agnostic learning sub-task* from ZS3. The *class-agnostic* task has a strong generalization to unseen categories, as demonstrated in [43,45]. In addition, as humans often associate semantics to whole images or at least segments, establishing a connection from semantics to *segment-level* visual features is more *natural* than that to *pixel-level* visual features. Therefore, our formulation is more efficient than *pixel-level zero-shot classification* in transferring knowledge from $S$ to $U$.

## 3.2. ZegFormer

Fig. 2 illustrates the pipeline of our proposed ZegFormer model. We first generate a set of segment-level embeddings and then project them for class-agnostic grouping and segment-level zero-shot classification by two parallel layers. A pre-trained image encoder is also used for segment classification. The two segment-level classification scores are finally fused to obtain the results.

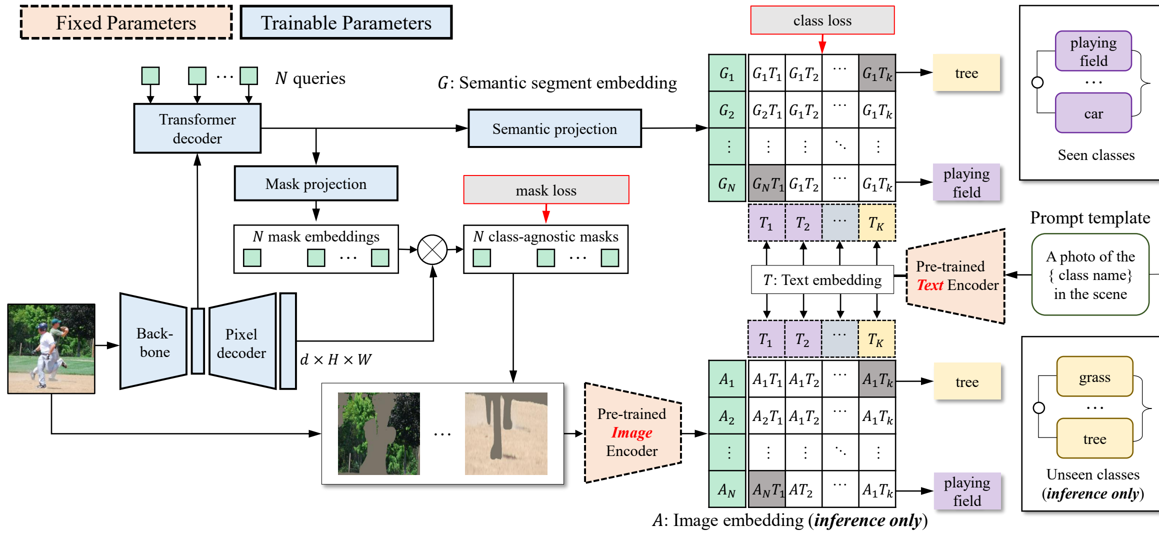


Figure 2. The pipeline of our proposed ZegFormer for zero-shot semantic segmentation. We first feed $N$ queries and feature maps to a transformer decoder to generate $N$ *segment embeddings*. We then feed each segment embedding to a mask projection layer and a semantic projection layer to obtain a mask embedding and a semantic embedding. Mask embedding is multiplied with the output of pixel decoder to generate a *class-agnostic binary mask*, while The semantic embedding is classified by the text embeddings. The text embeddings are generated by putting the class names into a prompt template and then feeding them to a text encoder of a vision-language model. During training, only the seen classes are used to train the segment-level classification head. During inference, both the text embeddings of *seen* and *unseen* classes are used for segment-level classification. We can obtain two segment-level classification scores with semantic segment embeddings and image embeddings. Finally, we fuse the these two classification scores as our final class prediction of segments.

**Segment Embeddings.** Recently, there are several segmentation models [12, 32, 59] that can generate a set of *segment-level* embeddings. We choose the Maskformer [12] as a basic semantic segmentation model for simplicity. By feeding $N$ segment queries and a feature map to a transformer decoder, we can obtain $N$ segment-level embeddings. Then we pass each segment embedding through a semantic projection layer and a mask projection layer to obtain a *mask embedding* and a *semantic embedding* for each segment. We denote segment-level *semantic embedding* (SSE) as $G_q \in \mathbb{R}^d$ and the segment-level *mask embedding* as $B_q \in \mathbb{R}^d$, where $q$ indexes a query.

**Class-Agnostic Grouping.** Denote the feature maps out from pixel decoder as $\mathcal{F}(I) \in \mathbb{R}^{d \times H \times W}$. The binary mask prediction for each query can be calculated as $m_q = \sigma(B_q \cdot \mathcal{F}(I)) \in [0, 1]^{H \times W}$, where $\sigma$ is the sigmoid function. Note that $N$ is usually smaller than the number of classes.

**Segment Classification with SSE.** For training and inference, each "class name" in a set of class names $C$ is put into a prompt template (*e.g.* "A photo of a {class name} in the scene") and then fed to a text encoder. Then we can obtain $|C|$ text embeddings, denoted as $T = \{T_c \in \mathbb{R}^d | c = 1, ...|C|\}$, where $C = S$ during training, while $C = S \cup U$ during inference. In our pipeline, we also need a "no object" category, if the intersection over union (IoU) between a segment and any ground truths is low. For the "no object" category, it is unreasonable to be presented by a single class name. Therefore, we add an extra learnable embedding $T_0 \in \mathbb{R}^d$ for "no object". The predicted probability distribution over the seen classes and the "no object" for a segment query is calculated as:

$$p_q(c) = \frac{\exp(\frac{1}{\tau} s_c(T_i, G_q))}{\sum_{i=0}^{|C|} \exp(\frac{1}{\tau} s_c(T_i, G_q))}, \tag{1}$$

where $q$ indexes a query. $s_c(\mathbf{e}, \mathbf{e}') = \frac{\mathbf{e} \cdot \mathbf{e}'}{|\mathbf{e}||\mathbf{e}'|}$ is the cosine similarity between two embeddings. $\tau$ is the temperature.

**Segment Classification with Image Embedding.** While the aforementioned steps can already form *a standalone approach* for ZS3, it is also possible to use an *image encoder* of a pre-trained vision-language model (*e.g.* CLIP [46]) to improve the classification accuracy on segments, owing to the flexibility of the decoupling formulation. In this module, we create a suitable sub-image for a segment. The process can be formulated as, given a mask prediction $m_q \in [0, 1]^{H \times W}$ for a query $q$, and the input image $I$, create a sub-image $I_q = f(m_q, I)$, where $f$ is a preprocess function (*e.g.*, a masked image or a cropped image according to $m_q$). We give detailed ablation studies in Sec. 4.5. We pass $I_q$ to a pre-trained image encoder and obtain image embedding $A_q$. Similar to Eq. 1, we can calculate a probability distribution, denoted as $p'_q(c)$.

**Training.** During the training of ZegFormer, only the pixel labels belonging to $S$ are used. To compute the training loss, a bipartite matching [9, 12] is performed between the predicted masks and the ground-truth masks. The loss of classification for each segment query is $-log(p_q(c_q^{gt}))$, where $c_q^{gt}$ belongs to $S$ if the segment is matched with a ground truth mask and "no object" if the segment does not have a matched ground truth segment. For a segment matched with ground truth segment $R_q^{gt}$, there is a mask loss $L_{mask}(m_q, R_q^{gt})$. In detail, we use the combination of a dice loss [40] and a focal loss [34].

**Inference.** During inference, we integrate the predicted binary masks and class scores of segments to obtain the final results of semantic segmentation. According to the class probability scores, we have three variants of ZegFormer.

(1) *ZegFormer-seg.* This variant use the segment classification scores with segment queries for inference by calculate a per-pixel class probability $\sum_{q=1}^{N} p_q(c) \cdot m_q[h,w]$, where $(h,w)$ is a location in an image. Since there is an imbalanced data problem, which results in predictions being biased to seen classes. Following [54], we calibrate the prediction by decreasing the scores of seen classes. The final category prediction for each pixel is then calculated as:

$$\underset{c \in S+U}{\arg\max} \sum_{q=1}^{N} p_q(c) \cdot m_q[h,w] - \gamma \cdot \mathbb{I}[c \in S], \quad (2)$$

where $\gamma \in [0,1]$ is the calibration factor. The indicator function $\mathbb{I}$ is equal to 1 when $c$ belongs to the seen classes.

(2) *ZegFormer-img.* The inference process of this variant is similar to Eq. 2. The only difference is that the $p_q(c)$ is replaced by $p_q^{'}(c)$.

(3) *ZegFormer.* This variant is our full model. We first fuse $p_q(c)$ and $p_q^{'}(c)$ for each query as:

$$p_{q,\text{fusion}}(c) = \begin{cases} p_q(c)^{\lambda} \cdot p_{q,\text{avg}}^{(1-\lambda)} & \text{if } c \in S \\ p_q(c)^{(1-\lambda)} \cdot p_q^{'}(c)^{\lambda} & \text{if } c \in U, \end{cases} \quad (3)$$

where a geometry mean of $p_q(c)$ and $p_q^{'}(c)$ will return, if $c \in U$. The contribution of the two classification scores is balanced by $\lambda$. Since $p_q(c)$ is usually more accurate than $p_q^{'}(c)$ if $c \in S$, we do not want $p_q^{'}(c)$ contribute to the prediction of $S$. Therefore, we calculate a geometry mean of $p_q(c)$ and $p_{q,\text{avg}} = \sum_{j \in S} p_q^{'}(j)/|S|$ on seen classes. This way, the probabilities of seen classes and unseen classes can be adjusted to the same range, and only $p_q(c)$ contributes to distinguishing seen classes. The final results for semantic segmentation are obtained by a process similar to Eq. 2.

# 4. Experiments

Since most of the previous works [7, 20, 42] focus on the GZS3 setting, we evaluate our method on the GZS3 setting in the main paper. See our results on the ZS3 setting in the appendix. We introduce the datasets and evaluation metrics that we use in the following.

## 4.1. Datasets and Evaluation Metrics

**COCO-Stuff** is a large-scale dataset for semantic segmentation that contains 171 valid classes in total. We use 118,287 training images as our training set and 5,000 validation images as our testing set. We follow the class split in [54] to choose 156 valid classes as seen classes and 15 valid classes as unseen classes. We also use a subset of the training classes as a validation set for tuning hyperparameters, following the cross-validation procedure [54, 55].

**PASCAL-VOC Dataset** has been split into 15 seen classes and 5 unseen classes in previous works [20, 54]. There are 10582 images for training and 1,449 images for testing.

**ADE20k-Full Dataset** is annotated in an open-vocabulary setting with more than 3,000 categories. It contains 25k images for training and 2k images for validation. We are the first that evaluate GZS3 methods on the challenging ADE20k-Full. Following the supervised setting in [12], we choose 847 classes that are present in both train and validation sets for evaluation, so that we can compare our ZegFormer with supervised models. We split the classes into seen and unseen according to their *frequency*. The seen classes are present in more than 10 images, while the unseen classes are present in less than 10 images. In this way, we obtain 572 classes for seen and 275 classes for unseen.

**Class-Related Segmentation Metric** includes the *mean intersection-over-union* (mIoU) averaged on seen classes, unseen classes, and their harmonic mean, which has been widely used in previous works of GZS3 [7, 54].

**Class-Agnostic Grouping Metric** has been well studied in [44]. We use the well-known *precision-recall for boundaries* ($P_b$, $R_b$, and $F_b$) as the evaluation metric, and use the public available code[1] for evaluation.

## 4.2. Implementation Details

Our implementation is based on Detectron2 [53]. For most of our ablation experiments, we use ResNet50 as our backbone and FPN [33] as the pixel decoder. When compared with the state-of-the-art methods, we use the ResNet-101 as our backbone. We use the text encoder and image encoder of ViT-B/16 CLIP model[2] in our implementation. We set the number of queries in the transformer decoder as 100 by default. The dimension of query embedding and transformer decoder is set as 256 by default. Since the dimension of text embeddings is 512, we use a projection layer to map the segment embeddings from 256 to 512 dimension. We empirically set the temperature $\tau$ in Eq. 1 to be 0.01. The image resolution of processed sub-images for segment classification with image-level embeddings is $224 \times 224$. For all the experiments, we use a batch size of 32. We train models for 60k and 10k iterations for COCO-Stuff and PASCAL

[1] https://github.com/jponttuset/seism
[2] https://github.com/openai/CLIP

Table 1. Comparisons with the baseline in class-related segmentation metric on COCO-stuff. We report the results with CLIP [46] text embeddings and the concatenation of fastText (ft) [27] and word2vec (w2v) [38] for each algorithm. We show the improvements from ft+w2v to CLIP text in brackets.

| | | Generalized zero shot | | |
|---|---|---|---|---|
| | cls. embed. | Seen | Unseen | Harmonic |
| SPNet-FPN | ft+w2v | 31.8 | 6.9 | 11.3 |
| | clip text | 32.3 (+0.5) | 11.0 (+4.1) | 16.4 (+5.1) |
| ZegFormer-seg (ours) | ft+w2v | 37.3 | 10.8 | 16.8 |
| | clip text | **37.4** (+0.1) | **21.4** (+10.6) | **27.2** (+10.4) |

VOC, respectively. We use the ADAMW as our optimizer with a learning rate of 0.0001 and 1e-4 weight decay.

### 4.3. Zero-Shot Pixel Classification Baseline

To compare *decoupling formulation* with *pixel-level zero-shot classification*, we choose SPNet [54] as our *pixel-level zero-shot classification* baseline, since SPNet and ZegFormer both belong to the *discriminative* methods with neat designs. For fair comparison, we reimplement the SPNet [54] with FPN (which is also used by ZegFormer). We denote this variant of SPNet as SPNet-FPN. We implement SPNet-FPN in the same codebase (*i.e.*, Detectron2 [53]) as our ZegFormer. The common settings are also the same as the ones used in ZegFormer).

### 4.4. Comparisons with the Baseline

**Class-Related Segmentation Metric.** We compare ZegFormer-seg with the baseline under two types of text embeddings (*i.e.* CLIP [46] text embeddings and the concatenation of fastText [27] (ft) and word2vec [38] (w2v).) The widely used ft + w2v in GZS3 [54] is trained only by *language data*. In contrast, CLIP text encoder is trained by multimodal contrastive learning of *vision and language*. From Tab. 1, we can conclude: 1) CLIP text embeddings [3] is better than the concatenation of fastText [27] and word2vec [38]; 2) Our proposed decoupling formulation for ZS3 is better than the commonly-used zero-short pixel classification ones, regardless of what class embedding methods are used; 3) *ZegFormer-seg has a much large gain than the SPNet-FPN*, when the class embedding method is changed from the ft + w2v to CLIP (**10.6** points *v.s.* **4.1** points improvements). We argue that the segment-level visual features are aligned better to the features of CLIP than the pixel-level visual features.

**Class-Agnostic Grouping Metric.** We compare the image grouping quality of ZegFormer-seg and the baseline on COCO-Stuff [8]. The image grouping quality is a *class-agnostic metric*, which can provide us more insight. We can see in Tab. 2 that ZegFormer significantly outperforms the baseline on the $F_b$, $P_b$, and $R_b$, regardless of what class

[3]Similar to works in object detection [19, 58], when using CLIP text embeddings, it is a relaxed setting compared to the pure GZS3 task. But this setting has a more realistic value. For simplicity, we still call it GZS3.

Table 2. Comparisons with the baseline in class-agnostic grouping metric on COCO-Stuff. In the brackets are the gaps to the SPNet-FPN (baseline).

| | cls. embed. | $F_b$ | $P_b$ | $R_b$ |
|---|---|---|---|---|
| SPNet-FPN | ft+w2v | 40.3 | 32.6 | 52.8 |
| ZegFormer-seg (ours) | ft+w2v | 49.4 (+9.1) | 43.5 (+10.9) | 57.2 (+4.4) |
| SPNet-FPN | clip text | 42.6 | 35.4 | 53.4 |
| ZegFormer-seg (ours) | clip text | 50.4 (+7.8) | 44.0 (+8.6) | 58.9 (+5.5) |

Table 3. Influence of preprocess of for sub images.

| | preprocess | Seen | Unseen | Harmonic |
|---|---|---|---|---|
| ZegFormer-seg | - | **37.4** | 21.4 | 27.2 |
| ZegFormer | crop | 36.6 | 19.7 | 25.6 |
| | mask | 36.0 | 31.0 | 33.3 |
| | crop and mask | 35.9 | **33.1** | **34.4** |

original image crop mask mask and crop

Figure 3. Comparison between three preprocess for a segment.

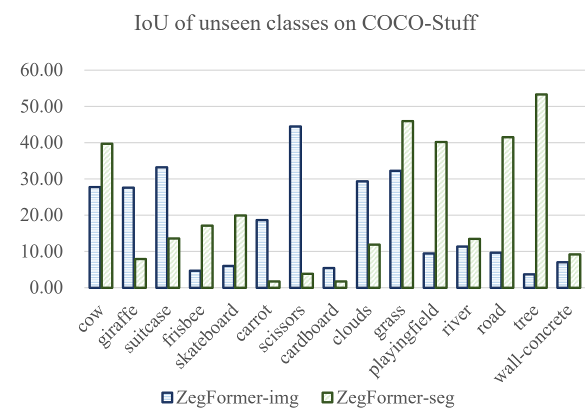


Figure 4. ZegFormer-seg *v.s.* ZegFormer-img in IoU of unseen classes on COCO-Stuff.

embedding methods are used. This verifies that the *decoupled ZS3 formulation* has much stronger generalization than *pixel-level zero-shot classification* to group the pixels of unseen classes.

### 4.5. Ablation Study on ZegFormer

**Preprocess for Sub-Images.** We explored three preprocess to obtain sub images (*i.e.* "crop", "mask", and "crop and mask") in the full model ZegFormer. The three ways to preprocess a segment for "tree" are shown in Fig. 3. We can see that when we merely crop a region from the original image, there may exist more than one categories, which will decrease the classification accuracy. In the masked image, the nuisances are removed, but there are many unnecessary pixels outside the segment. The combination of crop and mask can obtain a relatively suitable image for classification. We compare the influences of three ways for the

Table 4. Comparison with the previous GZS3 methods on PASCAL VOC and COCO-Stuff. The "Seen", "Unseen", and "Harmonic" denote mIoU of seen classes, unseen classes, and their harmonic mean. The STRICT [42] proposed a self-training strategy and applied it to SPNet [54]. The numbers of STRICT and SPNet (w/ self-training) are from [42]. Other numbers are from their original papers.

| | | | | PASCAL VOC | | | COCO-Stuff | | |
|---|---|---|---|---|---|---|---|---|---|
| Methods | Type | w/ Self-train. | Re-train. for new classes? | Seen | Unseen | Harmonic | Seen | Unseen | Harmonic |
| SPNet [54] | discriminative | ✗ | ✗ | 78.0 | 15.6 | 26.1 | 35.2 | 8.7 | 14.0 |
| ZS3 [7] | generative | ✗ | ✓ | 77.3 | 17.7 | 28.7 | 34.7 | 9.5 | 15.0 |
| CaGNet [20] | generative | ✗ | ✓ | 78.4 | 26.6 | 39.7 | 33.5 | 12.2 | 18.2 |
| SIGN [13] | generative | ✗ | ✓ | 75.4 | 28.9 | 41.7 | 32.3 | 15.5 | 20.9 |
| Joint [5] | discriminative | ✗ | ✗ | 77.7 | 32.5 | 45.9 | - | - | - |
| ZS3 [7] | generative | ✓ | ✓ | 78.0 | 21.2 | 33.3 | 34.9 | 10.6 | 16.2 |
| CaGNet [20] | generative | ✓ | ✓ | 78.6 | 30.3 | 43.7 | 35.6 | 13.4 | 19.5 |
| SIGN [13] | generative | ✓ | ✓ | 83.5 | 41.3 | 55.3 | 31.9 | 17.5 | 22.6 |
| SPNet [42] | discriminative | ✓ | ✓ | 77.8 | 25.8 | 38.8 | 34.6 | 26.9 | 30.3 |
| STRICT [42] | discriminative | ✓ | ✓ | 82.7 | 35.6 | 49.8 | 35.3 | 30.3 | 32.6 |
| ZegFormer | discriminative | ✗ | ✗ | **86.4** | **63.6** | **73.3** | **36.6** | **33.2** | **34.8** |

ZegFormer in Tab. 3. We can see that the combination of crop and mask can get the best performance, while only using the crop preprocess is lower than the performance of ZegFormer-seg, which does not use an image embedding for segment classification.

**ZegFormer-seg *v.s.* ZegFormer-img.** We compare their performance for unseen categories on COCO-Stuff in Fig. 4. We can see that the ZegFormer-img is better at things categories (such as "giraffe", "suitacase", and "carrot", etc), while worse at stuff categories (such as "grass", "playingfield", and "river", etc.) Therefore, these two kinds of classification scores are complementary, which illustrates why their fusion will improve performance.

### 4.6. Comparison with the State-of-the-art

We compare our ZegFormer with the previous methods in Tab. 4. Specifically, ZegFormer outperforms Joint [5] by **31 points** in the mIoU of unseen classes on PASCAL VOC [15], and outperforms SIGN [13] by **18 points** in the mIoU of unseen classes on COCO-Stuff [8]. When compared to the results with *self-training*, ZegFormer outperforms SIGN [13] by **22 points** in mIoU of unseen classes on PASCAL VOC [15], and STRICT [42] by **3 points** on COCO-Stuff [8]. It is worth noting that the *generative* methods and *self-training* methods need a complicated multi-stage training scheme. They also need to be *re-trained* whenever new classes are incoming (although semantic category labels of unseen classes are not required). The self-training also needs to access the unlabelled pixels of unseen classes to generate pseudo labels. In contrast, our ZegFormer is a *discriminative* method, which is much simpler than those methods and can be applied to any unseen classes *on-the-fly*. Similar to our ZegFormer, SPNet [54] (without self-training) and Joint [5] are also discriminative methods that can be flexibly applied to unseen classes, but their performances are much worse than ours.

Table 5. Results on ADE20k-Full. Our ZegFormer is comparable with a fully supervised model.

| | Backbone | Seen | Unseen | Harmonic |
|---|---|---|---|---|
| SPNet-FPN | R-50 | 9.2 | 0.9 | 1.6 |
| ZegFormer (ours) | R-50 | **17.4** | **5.3** | **8.1** |
| Fully supervised | R-50 | 19.7 | 5.6 | 8.7 |

### 4.7. Results on ADE20k-Full

Since we are the first that report the results on the challenging ADE20k-Full [64] for GZS3, there are no other methods for comparison. We compared ZegFormer with SPNet-FPN, and a fully supervised Maskformer trained on both seen and unseen classes. From Tab. 5, we can see that our ZegFormer significantly surpasses the SPNet-FPN baseline by **4 points** in mIoU unseen and is *even comparable with the fully supervised model*. We can also see that the dataset is very challenging that even the supervised model only achieved 5.6 points in the mIoU unseen, which indicates there is still much room for improvements.

## 5. Visualization

For the visualization analyses, we use a SPNet-FPN and a ZegFormer-seg trained with **156 seen classes** on COCO-Stuff.Then we test the two models with different sets of class names. **Results with 171 classes from COCO-Stuff.** From Fig. 5, we can see that when segmenting unseen classes, it is usually confused by similar classes. Since the pixels of a segment have large variations, these pixel-level classifications are not consistent. In contrast, decoupling formulation can obtain high-quality segmentation results. Based on the segmentation results, the segment-level zero-shot classification is more related to CLIP pretraining. Therefore, ZegFormer-seg can obtain more accurate classification results.

**Results with 847 classes from ADE20k-Full.** As shown in Fig. 6, the SPNet-FPN (pixel-level zero-shot classification) is much worse than the ZegFormer-seg (our decoupling formulation). The reason is that there is severe com-

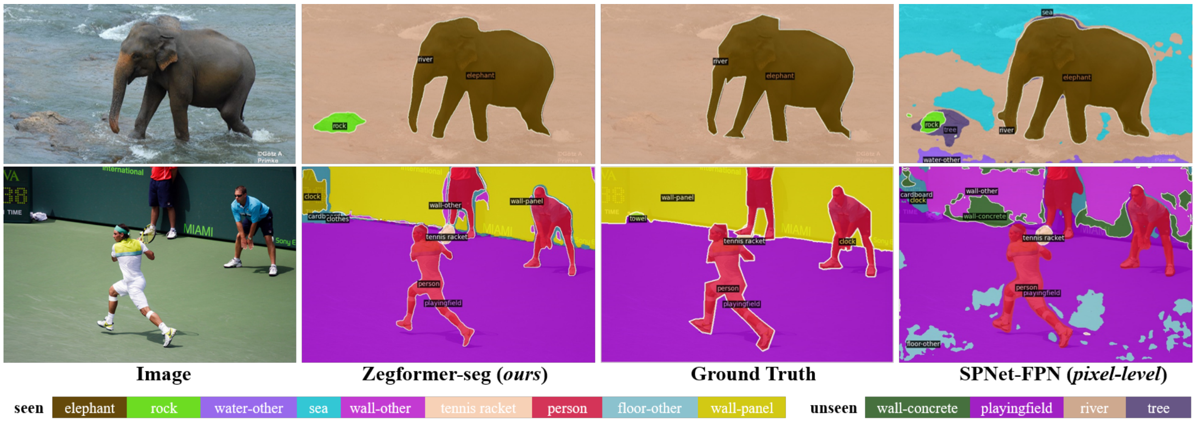


Figure 5. Results on COCO-Stuff, using 171 class names in COCO-Stuff to generate text embeddings. ZegFormer-seg (our decoupling formulation of ZS3) is better than SPNet-FPN (pixel-level zero-shot classification) in segmenting unseen categories.

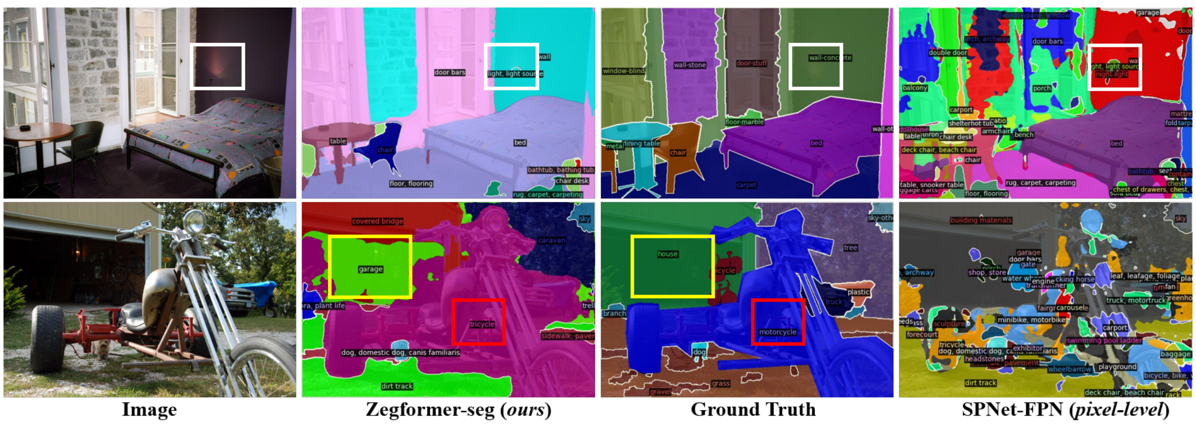


Figure 6. Results on COCO-Stuff, using 847 class names from ADE20k-Full to generate text embeddings. We can see that the SPNet-FPN (*pixel-level zero-shot classification baseline*) is very unstable when the number of unseen classes is large. We can also see that the set of 847 class names provide richer information than the set of 171 class names in COCO-Stuff. For example, in the yellow box of the second row, the predicted "garage" for a segment is labeled as "house" in COCO-Stuff. We provide more visualization results in our appendix.

petition among classes at pixel-level classification when we use 847 classes for inference. In contrast, the decoupling model is not influenced by the number of classes. These results confirm that *the decoupling formulation is a right way to achieve human-level ability to segment objects with a large number of unseen classes*. We can also see that the set of 847 class names contains richer information to generate text embeddings for inference than the set of 171 class names. For example, the unannotated "light, light source" is segmented by the two ZS3 models (white box in the $1_{st}$ row of Fig. 6.) The labeled "motorcycle" is predicted as "tricycle" by ZegFormer (red box in the $2_{nd}$ row of Fig. 6).

## 6. Conclusion

We reformulate the ZS3 task by decoupling. Based on the new formulation, we propose a simple and effective ZegFormer for the task of ZS3, which demonstrates significant advantages to the previous works. The proposed ZegFormer provides a new manner to study the ZS3 problem and serves as a strong baseline. Beyond the ZS3 task, ZegFormer also has the potentials to be used for few-shot semantic segmentation. We leave it for further research.

**Limitations.** Although the full model of ZegFormer shows superiority in all the situations, we empirically find that ZegFormer-seg does not perform well when the scale of training data is small. One possible reason is that the transformer structure needs a large number of data for training, which has been discussed in [35]. A more efficient training strategy for ZegFormer-seg or using other mask classification methods such as K-Net [59] may alleviate this issue, and can be studied in the future.

**Acknowledgement.** This work was supported by National Nature Science Foundation of China under grant 61922065, 62101390, 41820104006 and 61871299. The numerical calculations in this paper have been done on the supercomputing system in the Supercomputing Center of Wuhan University. Jian Ding was also supported by the China Scholarship Council. We would like to thank Tianzhu Xiang, Di Hu, XingXing Weng and Zeran Ke for insightful discussion.

## References


[1] Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. Evaluation of output embeddings for fine-grained image classification. In *CVPR*, pages 2927–2936, 2015. 1

[2] Ziad Al-Halah, Makarand Tapaswi, and Rainer Stiefelhagen. Recovering the missing link: Predicting class-attribute associations for unsupervised zero-shot learning. In *CVPR*, pages 5975–5984, 2016. 2

[3] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE TPAMI*, 33(5):898–916, 2010. 3

[4] Pablo Arbeláez, Jordi Pont-Tuset, Jonathan T Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping. In *CVPR*, pages 328–335, 2014. 3

[5] Donghyeon Baek, Youngmin Oh, and Bumsub Ham. Exploiting a joint embedding space for generalized zero-shot semantic segmentation. *arXiv:2108.06536*, 2021. 3, 7

[6] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987. 1

[7] Maxime Bucher, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Zero-shot semantic segmentation. *NeurIPS*, 32:468–479, 2019. 1, 3, 5, 7

[8] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *CVPR*, pages 1209–1218, 2018. 2, 6, 7

[9] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020. 4

[10] 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 TPAMI*, 40(4):834–848, 2018. 1

[11] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 801–818, 2018. 1

[12] Bowen Cheng, Alexander G Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *NeurIPS*, 2021. 2, 4, 5

[13] Jiaxin Cheng, Soumyaroop Nandi, Prem Natarajan, and Wael Abd-Almageed. Sign: Spatial-information incorporated generative network for generalized zero-shot semantic segmentation. *arXiv:2108.12517*, 2021. 7

[14] Yining Deng, B Shin Manjunath, and Hyundoo Shin. Color image segmentation. In *CVPR*, volume 2, pages 446–451. IEEE, 1999. 3

[15] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 2, 7

[16] Rob Fergus, Hector Bernal, Yair Weiss, and Antonio Torralba. Semantic label sharing for learning with many categories. In *ECCV*, pages 762–775. Springer, 2010. 2

[17] Andrea Frome, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomás Mikolov. Devise: A deep visual-semantic embedding model. In *NeurIPS*, 2013. 2

[18] Yanwei Fu, Tao Xiang, Yu-Gang Jiang, Xiangyang Xue, Leonid Sigal, and Shaogang Gong. Recent advances in zero-shot recognition. *arXiv:1710.04837*, 2017. 1

[19] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Zero-shot detection via vision and language knowledge distillation. *arXiv:2104.13921*, 2021. 2, 6

[20] Zhangxuan Gu, Siyuan Zhou, Li Niu, Zihan Zhao, and Liqing Zhang. Context-aware feature generation for zero-shot semantic segmentation. In *ACM MM*, pages 1921–1929, 2020. 3, 5, 7

[21] 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:2009.12232*, 2020. 3

[22] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 3

[23] Ronghang Hu, Piotr Dollár, Kaiming He, Trevor Darrell, and Ross Girshick. Learning to segment every thing. In *CVPR*, pages 4233–4241, 2018. 3

[24] Dinesh Jayaraman and Kristen Grauman. Zero shot recognition with unreliable attributes. *arXiv:1409.4327*, 2014. 1, 2

[25] Jingwei Ji, Shyamal Buch, Alvaro Soto, and Juan Carlos Niebles. End-to-end joint semantic segmentation of actors and actions in video. In *ECCV*, pages 702–717, 2018. 3

[26] 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:2102.05918*, 2021. 2

[27] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. Fasttext. zip: Compressing text classification models. *arXiv:1612.03651*, 2016. 1, 3, 6

[28] Pichai Kankuekul, Aram Kawewong, Sirinart Tangruamsub, and Osamu Hasegawa. Online incremental attribute-based zero-shot learning. In *CVPR*, pages 3657–3664. IEEE, 2012. 2

[29] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *arXiv:1703.04977*, 2017. 3

[30] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE TPAMI*, 36(3):453–465, 2013. 1, 2

[31] Peike Li, Yunchao Wei, and Yi Yang. Consistent structural relation learning for zero-shot segmentation. *NeurIPS*, 33, 2020. 3

[32] Zhiqi Li, Wenhai Wang, Enze Xie, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, Tong Lu, and Ping Luo. Panoptic segformer. *arXiv:2109.03814*, 2021. 2, 4

[33] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 5

[34] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 5

[35] Yahui Liu, Enver Sangineto, Wei Bi, Nicu Sebe, Bruno Lepri, and Marco De Nadai. Efficient training of visual transformers with small-size datasets. *arXiv:2106.03746*, 2021. 8

[36] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 1

[37] Thomas Mensink, Efstratios Gavves, and Cees GM Snoek. Costa: Co-occurrence statistics for zero-shot classification. In *CVPR*, pages 2441–2448, 2014. 2

[38] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, pages 3111–3119, 2013. 3, 6

[39] George A Miller. Wordnet: a lexical database for english. *CACM*, 38(11):39–41, 1995. 2

[40] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, pages 565–571. IEEE, 2016. 5

[41] Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S Corrado, and Jeffrey Dean. Zero-shot learning by convex combination of semantic embeddings. *arXiv:1312.5650*, 2013. 2

[42] Giuseppe Pastore, Fabio Cermelli, Yongqin Xian, Massimiliano Mancini, Zeynep Akata, and Barbara Caputo. A closer look at self-training for zero-label semantic segmentation. In *CVPRW*, pages 2693–2702, 2021. 3, 5, 7

[43] Pedro O Pinheiro, Ronan Collobert, and Piotr Dollár. Learning to segment object candidates. *arXiv:1506.06204*, 2015. 3

[44] Jordi Pont-Tuset and Ferran Marques. Supervised evaluation of image segmentation and object proposal techniques. *IEEE TPAMI*, 38(7):1465–1478, 2015. 2, 3, 5

[45] Lu Qi, Jason Kuen, Yi Wang, Jiuxiang Gu, Hengshuang Zhao, Zhe Lin, Philip Torr, and Jiaya Jia. Open-world entity segmentation. *arXiv:2107.14228*, 2021. 3

[46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021. 2, 4, 6

[47] Marcus Rohrbach, Michael Stark, and Bernt Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *CVPR*, pages 1641–1648. IEEE, 2011. 1, 2

[48] Marcus Rohrbach, Michael Stark, György Szarvas, Iryna Gurevych, and Bernt Schiele. What helps where–and why? semantic relatedness for knowledge transfer. In *CVPR*, pages 910–917. IEEE, 2010. 2

[49] Feihong Shen, Jun Liu, and Ping Hu. Conterfactual generative zero-shot semantic segmentation. *arXiv:2106.06360*, 2021. 3

[50] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE TPAMI*, 22(8):888–905, 2000. 2, 3

[51] 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 *ACM MM*, pages 4125–4134, 2020. 3

[52] Zhuowen Tu and Song-Chun Zhu. Image segmentation by data-driven markov chain monte carlo. *IEEE TPAMI*, 24(5):657–673, 2002. 2, 3

[53] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 5, 6

[54] Yongqin Xian, Subhabrata Choudhury, Yang He, Bernt Schiele, and Zeynep Akata. Semantic projection network for zero-and few-label semantic segmentation. In *CVPR*, pages 8256–8265, 2019. 1, 3, 5, 6, 7

[55] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE TPAMI*, 41(9):2251–2265, 2018. 5

[56] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *NeurIPS*, 2021. 1

[57] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *ECCV*, pages 173–190. Springer, 2020. 1

[58] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *CVPR*, pages 14393–14402, 2021. 6

[59] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. K-net: Towards unified image segmentation. *arXiv:2106.14855*, 2021. 2, 4, 8

[60] Ziming Zhang and Venkatesh Saligrama. Zero-shot learning via joint latent similarity embedding. In *CVPR*, pages 6034–6042, 2016. 1, 2

[61] Hang Zhao, Xavier Puig, Bolei Zhou, Sanja Fidler, and Antonio Torralba. Open vocabulary scene parsing. In *ICCV*, pages 2002–2010, 2017. 3

[62] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pages 2881–2890, 2017. 1

[63] Ye Zheng, Jiahong Wu, Yongqiang Qin, Faen Zhang, and Li Cui. Zero-shot instance segmentation. In *CVPR*, pages 2593–2602, 2021. 3

[64] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, pages 633–641, 2017. 2, 7

[65] Song Chun Zhu and Alan Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE TPAMI*, 18(9):884–900, 1996. 3