SED: A Simple Encoder-Decoder for Open-Vocabulary Semantic Segmentation

Bin Xie[1], Jiale Cao[1]*, Jin Xie[2], Fahad Shahbaz Khan[3,4], and Yanwei Pang[1,5]
[1]Tianjin University [2]Chongqing University
[3]Mohamed bin Zayed University of Artificial Intelligence
[4]Linköping University [5]Shanghai Artificial Intelligence Laboratory
{bin_xie,connor,pyw}@tju.edu.cn xiejin@cqu.edu.cn fahad.khan@mbzuai.ac.ae

Abstract

Open-vocabulary semantic segmentation strives to distinguish pixels into different semantic groups from an open set of categories. Most existing methods explore utilizing pre-trained vision-language models, in which the key is to adapt the image-level model for pixel-level segmentation task. In this paper, we propose a simple encoder-decoder, named SED, for open-vocabulary semantic segmentation, which comprises a hierarchical encoder-based cost map generation and a gradual fusion decoder with category early rejection. The hierarchical encoder-based cost map generation employs hierarchical backbone, instead of plain transformer, to predict pixel-level image-text cost map. Compared to plain transformer, hierarchical backbone better captures local spatial information and has linear computational complexity with respect to input size. Our gradual fusion decoder employs a top-down structure to combine cost map and the feature maps of different backbone levels for segmentation. To accelerate inference speed, we introduce a category early rejection scheme in the decoder that rejects many no-existing categories at the early layer of decoder, resulting in at most 4.7 times acceleration without accuracy degradation. Experiments are performed on multiple open-vocabulary semantic segmentation datasets, which demonstrates the efficacy of our SED method. When using ConvNeXt-B, our SED method achieves mIoU score of 31.6% on ADE20K with 150 categories at 82 millisecond (ms) per image on a single A6000. Our source code is available at `https://github.com/xb534/SED`.

1. Introduction

Semantic segmentation aims to parse semantic categories of each pixel in an image. Traditional methods [6, 31, 52] assume that the semantic categories are closed-set and struggle to recognize unseen category during inference. To this end, recent works have explored open-vocabulary semantic segmentation [3, 51, 59] to segment the pixels belonging to the arbitrary categories.

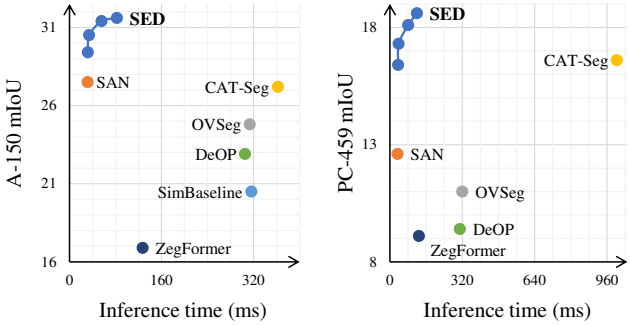

Figure 1. **Accuracy (mIoU) and speed (ms) comparison** on `A-150` and `PC-459`. Here, the speed is reported on a single NVIDIA A6000 GPU, and the changing factor of different points is the input size of images. Our SED achieves an optimal trade-off in terms of speed and accuracy compared to existing methods in literature: SAN [56], CAT-Seg [12], OVSeg [29], DeOP [22], SimBaseline [55] and ZegFormer [14].

Recently, vision-language models, such as CLIP [38] and ALIGN [25], learn aligned image-text feature representation from millions of image-text paired data. The pre-trained vision-language models exhibit superior generalization ability to recognize open-vocabulary categories. This motivates a body of research works to explore using vision-language models for open-vocabulary semantic segmentation [14, 29]. Initially, research works mainly adopt two-stage framework [15, 29, 55] to directly adapt vision-language models for open-vocabulary segmentation. Specifically, they first generate class-agnostic mask proposals and then adopt the pre-trained vision-language models to classify these proposals into different categories. However, such a two-stage framework uses two independent networks for mask generation and classification, thereby hampering computational efficiency. Further, it does not fully utilize contextual information.

Different to aforementioned two-stage approaches, methods based on single-stage framework directly extend a

*Corresponding author: Jiale Cao

single vision-language model for open-vocabulary segmentation. Several methods remove pooling operation in last layer of image encoder and generate pixel-level feature map for segmentation. For instance, MaskCLIP [62] removes global pooling at last layer of CLIP image encoder and uses the value-embeddings and text-embeddings to directly predict pixel-level segmentation map. CAT-Seg [12] first generates pixel-level image-text cost map and then refines the cost map with spatial and class aggregation. While these approaches achieve favorable performance, we note their following limitations. First, both MaskCLIP and CAT-Seg employ plain transformer ViT [16] as backbone which suffers from weak local spatial information and low-resolution input size. To address those issues, CAT-Seg introduces an additional network to provide spatial information. However, this incurs extra computational cost. Second, the computational cost of CAT-Seg significantly increases with the larger number of categories.

To address the issues above, we propose a simple yet effective encoder-decoder approach, named SED. Our SED comprises a hierarchical encoder-based cost map generation and a gradual fusion decoder with category early rejection. The hierarchical encoder-based cost map generation employs hierarchical backbone, instead of plain transformer, to predict pixel-level image-text cost map. Compared to plain transformer, hierarchical backbone better preserves the spatial information at different levels and has a linear computational complexity with respect to the input size. Our gradual fusion decoder gradually combines the feature maps from different levels of hierarchical backbone and cost map for segmentation prediction. To increase inference speed, we design a category early rejection scheme in decoder that effectively predicts existing categories and rejects non-existing categories at early layer of decoder. Comprehensive experiments are conducted on multiple open-vocabulary semantic segmentation datasets, revealing the merits of proposed contributions in terms of accuracy and efficiency. We summarize the contributions as follows.

- We propose an encoder-decoder for open-vocabulary semantic segmentation comprising a hierarchical encoder-based cost map generation and a gradual fusion decoder.
- We introduce a category early rejection scheme to reject non-existing categories at the early layer, which aids in markedly increasing inference speed without any significant degradation in segmentation performance. For instance, it provides 4.7 times acceleration on PC-459.
- Our proposed method, SED, achieves the superior performance on multiple open-vocabulary segmentation datasets. Specifically, the proposed SED provides a good trade-off in terms of segmentation performance and speed (see Fig. 1). When using ConvNeXt-L, our proposed SED obtains mIoU scores of 35.2% on `A-150` and 22.6% on `PC-459`.

2. Related Work

2.1. Semantic Segmentation

Traditional semantic segmentation methods mainly contain FCN-based approaches and transformer-based approaches. Initially, the researchers focused on FCN-based approaches. Long *et al.* [31] proposed one of the earliest fully-convolutional networks that fuses both deep and shallow features for improved segmentation. Afterwards, many FCN-based variants were proposed. Some methods utilize spatial pyramid network [6, 57] or encoder-decoder structure [1, 2, 5, 40, 48, 49] to extract local contextual information. Some methods [18, 24] exploit using attention module to extract non-local contextual information. Recently, the researchers focused on developing transformer-based approaches. Some methods [8, 20, 32, 52] employ the transformer as backbone to extract deep features, while some other methods [9, 10, 43, 60] employ transformer design as a segmentation decoder.

2.2. Vision-Language Models

Vision-language models learn the connection between image representation and text embeddings. Initially, the researchers developed vision-language models [7, 33, 44, 50] based on the pre-trained visual and language models, and explored to jointly fine-tune them on different downstream tasks. In contrast, CLIP [38] collects a large-scale image-text paired data from website and learns visual features via language supervision from scratch. The learned CLIP on large-scale data has a superior performance on different zero-shot tasks. Instead of using cleaned image-text paired data, ALIGN [25] learns visual-language representation from noisy image-text dataset. To achieve this goal, ALIGN employs a dual-encoder structure with contrastive loss, which achieves a good zero-shot performance on downstream tasks. Recently, Cherti *et al.* [11] conducted deep analysis on contrastive language-vision learning. Schuhmann *et al.* [41] built a billion image-text paired dataset for training large-scale vision-language models.

2.3. Open-Vocabulary Semantic Segmentation

Open-vocabulary semantic segmentation aims at segmenting arbitrary categories. Initially, the researchers [3, 51, 59] explored to align visual features with pre-trained text embeddings via a learned feature mapping. With the success of large-scale vision-language model CLIP [38], the researchers started to explore open-vocabulary semantic segmentation using CLIP. Some methods [14, 55] adopt two-stage framework that first predicts class-agnostic mask proposals and second classifies these proposals into different categories. To improve classification performance at second stage, OVSeg [29] fine-tunes the CLIP model on the masked image and their text annotations. Ding *et al.* [15] integrated

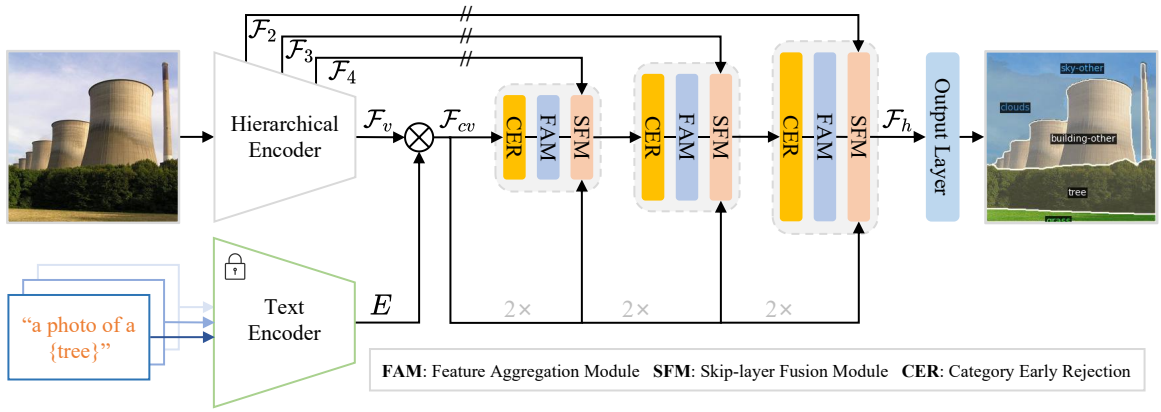

Figure 2. **Overall architecture of our proposed SED.** // represents stopping gradient back-propagation, $\otimes$ indicates cosine similarity, and $2\times$ indicates two times upsampling. We first employ hierarchical encoder (learnable) and text encoder (frozen) to generate pixel-level image-text cost map. Afterwards, we introduce a gradual fusion decoder to combine different feature maps of hierarchical encoder and cost map. The gradual fusion decoder stacks feature aggregation module (FAM) and skip-layer fusion module (SFM). In addition, we design a category early rejection (CER) in decoder which only used during inference, to accelerate speed without sacrificing performance.

mask tokens with pre-trained CLIP model for mask refinement and classification. ODISE [54] employs text-to-image diffusion model to generate mask proposals and perform classification. To enhance open-vocabulary performance, ODISE [54] further performs mask classification using the features cropped from pre-trained CLIP.

In contrast, some methods adopt single-stage framework. LSeg [28] learns pixel-level image features guided by the pre-trained CLIP text embeddings. MaskCLIP [62] removes the self-attention pooling layer to generate pixel-level feature map and employs text-embeddings to predict final segmentation map. SAN [56] introduces a side adapter network along the frozen CLIP model to perform mask prediction and classification. FC-CLIP [58] employs a frozen convolutional CLIP to predict class-agnostic masks and employs mask-pooled features for classification. CAT-Seg [12] generates pixel-level cost map and refines the cost map for segmentation prediction. Our proposed method is inspired by CAT-Seg in which fine-tuning image encoder through cost map does not degrade its open-vocabulary ability, but has significant differences: (1) Our SED is a simpler framework without additional backbone, and has a better performance and faster inference speed. (2) Our SED employs hierarchical image encoder to generate cost map and to perform skip-layer fusion, which can significantly improve performance and has linear computational cost with respect to the input size. (3) In decoder, we introduce a simple large-kernel operation and gradual fusion for feature aggregation, and design a category early rejection strategy for acceleration without sacrificing performance.

3. Method

In this section, we describe our proposed encoder-decoder for open-vocabulary semantic segmentation, named SED. Fig. 2 shows the overall architecture of our proposed SED, which comprises two main components: a hierarchical encoder-based cost map generation and a gradual fusion decoder with category early rejection. In our hierarchical encoder-based cost map generation, we employ hierarchical image encoder and text encoder to generate pixel-level image-text cost map $\mathcal{F}_{cv}$ and hierarchical feature maps $\mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_4$ for the decoder. Our gradual fusion decoder employs feature aggregation module (FAM) and skip-layer fusion module (SFM) to gradually combine pixel-level cost map $\mathcal{F}_{cv}$ and hierarchical feature maps $\mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_4$ for generating high-resolution feature map $\mathcal{F}_h$. Based on $\mathcal{F}_h$, we employ an output layer to predict segmentation maps of different categories. In addition, a category early rejection (CER) strategy is used in the decoder to early reject non-existing categories for boosting inference speed.

3.1. Hierarchical Encoder-based Cost Map

Hierarchical encoder-based cost map generation (HECG) adopts the vision-language model CLIP [11, 38, 41] to generate pixel-level image-text cost map. Specifically, we first employ hierarchical image encoder and a text encoder to respectively extract visual features and text embeddings. Then, we calculate pixel-level cost map between these two features. Existing methods such as MaskCLIP [62] and CAT-Seg [12] adopt the plain transformer as image encoder to generate pixel-level cost map. As discussed earlier, plain transformer suffers from relatively weak local spatial information and has quadratic complexity with respect to the input size. To address those issues, we propose to use hierarchical backbone as image encoder for cost map generation. Hierarchical encoder can better capture local information and has linear complexity with respect to the input size. The cost map generation is described as follow.

Given an input image $I \in \mathbb{R}^{H\times W\times 3}$, we first utilize a

hierarchical encoder ConvNeXt [30, 41] to extract multi-scale feature maps, denoted as $\mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_4, \mathcal{F}_5$. These feature maps have strides of $4, 8, 16, 32$ pixels with respect to the input size. To align the output visual features and text embeddings, an MLP layer is attached at the last feature map $\mathcal{F}_5$ to obtain an aligned visual feature map $\mathcal{F}_v \in \mathbb{R}^{H_v \times W_v \times D_t}$, where D_t is equal to the feature dimension of text embeddings, H_v is $H/32$, and W_v is $W/32$. Given an arbitrary set of category names $\{T_1, .., T_N\}$, we use the prompt template strategy [12, 21, 29] to generate different textual descriptions $S(n) \in \mathbb{R}^P$ about category name T_n, such as "a photo of a $\{T_n\}$, a photo of many $\{T_n\}$, ...". N represents the total number of categories, and P is the number of templates for each category. By fed $S(n)$ to the text encoder, we obtain text embeddings, denoted as $E = \{E_1, .., E_N\} \in \mathbb{R}^{N \times P \times D_t}$. By calculating the cosine similarity [39] between visual feature map $\mathcal{F}_v$ and text embeddings E, we obtain the pixel-level cost map $\mathcal{F}_{cv}$ as

$$\mathcal{F}_{cv}(i, j, n, p) = \frac{\mathcal{F}_v(i, j) \cdot E(n, p)}{\|\mathcal{F}_v(i, j)\| \|E(n, p)\|}, \tag{1}$$

where i, j indicate the 2D spatial position, n represents the index of text embeddings, and p represents the index of templates. Therefore, the initial cost map $\mathcal{F}_{cv}$ has the size of $H_v \times W_v \times N \times P$. The initial cost map goes through a convolutional layer to generate the input feature map $\mathcal{F}^{l1}_{dec} \in \mathbb{R}^{H_v \times W_v \times N \times D}$ of the decoder. For simplicity, we do not show $\mathcal{F}^{l1}_{dec}$ in Fig. 2.

3.2. Gradual Fusion Decoder

Semantic segmentation greatly benefits from high-resolution feature maps. However, the cost map $\mathcal{F}_{cv}$ generated by encoder has relatively low resolution and noisy. Therefore, it is not beneficial to generate high-quality segmentation map by directly using cost map for prediction. To address this issue, we propose a gradual fusion decoder (GFD). GFD gradually generates high-resolution feature map $\mathcal{F}_h$ by cascading two modules, including feature aggregation module (FAM) and skip-layer fusion module (SFM), into multiple layers. FAM aims to model the relationship between local regions and different classes, whereas SFM is designed to enhance the local details of feature maps using shallow features of hierarchical encoder.

Feature Aggregation Module: Fig. 3(a) shows the design of feature aggregation module (FAM) that has spatial-level and class-level fusion. We first perform spatial-level fusion to model the relationship of local region. Prior works [30, 36] have demonstrated that large-kernel convolutional operation is a simple but efficient structure to capture local information. Motivated by this, we perform spatial-level fusion employing large-kernel convolution [30]. Specifically, the input feature map $\mathcal{F}^{li}_{dec}$ goes through a depth-wise convolutional layer and an MLP layer.

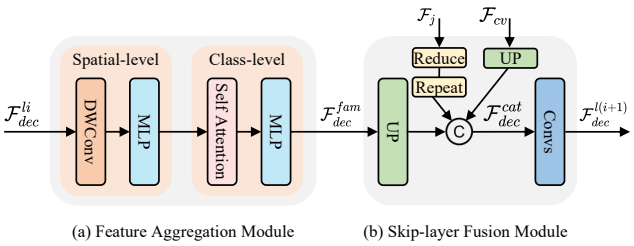

Figure 3. **Structure of gradual fusion decoder.** The gradual fusion decoder (GFD) first performs feature aggregation (a) in both spatial and class levels, and then employs skip-layer fusion (b) to combine the feature maps from both previous decoder layer and hierarchical encoder.

The depth-wise convolutional layer has a 9×9 depth-wise convolution and a layer-norm operation, and the MLP layer contains two linear layers and a GeLU layer. In addition, we use a residual connection in both convolutional and MLP layers. Following the spatial-level aggregation, we further apply a linear self-attention operation as in [12, 26] along category dimension to perform class-level feature aggregation. The generated feature map by feature aggregation module (FAM) is represented as $\mathcal{F}^{fam}_{dec}$.

Skip-layer Fusion Module: The feature map $\mathcal{F}^{fam}_{dec}$ is spatially coarser, which lacks local detail information. In contrast, the shallow feature maps $\mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_4$ in hierarchical encoder contains rich detail information. To incorporate these local details for segmentation, we introduce the skip-layer fusion module to gradually combine the low-resolution feature map $\mathcal{F}^{fam}_{dec}$ with high-resolution feature maps $\mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_4$. As shown in Fig. 3(b), we first upsample low-resolution feature map $\mathcal{F}^{fam}_{dec}$ by a factor of 2 using the deconvolutional operation. Then, we reduce the channel dimension of the corresponding high-resolution feature map $\mathcal{F}_j, j \in 2, 3, 4$ by a factor of 16 using the convolutional operation, and repeat the reduced feature map N times to have the same category dimension with $\mathcal{F}^{fam}_{dec}$. Afterwards, we concatenate the upsampled feature map and the repeated feature map together. To fuse more information, we also upsample and concatenate the initial cost map $\mathcal{F}_{cv}$. Finally, we feed the concatenated feature map $\mathcal{F}^{cat}_{dec}$ through two convolutional layers to generate the output feature map $\mathcal{F}^{l(i+1)}_{dec}$. As observed in [12], directly back-propagating the gradient to the image encoder degrades the performance of open-vocabulary semantic segmentation. Therefore, we stop gradient back-propagation directly from skip-layer fusion module to the image encoder.

Compared to plain transformer, hierarchical encoder with skip-layer fusion significantly improves the performance. This is likely due to that, the hierarchical encoder is able to provide rich local information for segmentation, and the stopped gradient back-propagation avoids the negative impact on open-vocabulary segmentation ability.

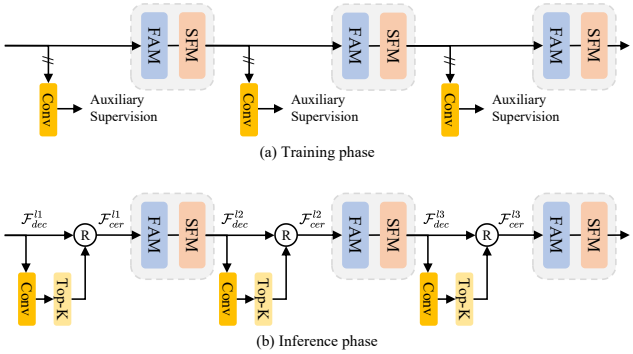

Figure 4. **Structure of category early rejection.** Circle with R represents removing feature maps of non-selected categories. During training (a), we attach an auxiliary convolution after decoder layer to predict segmentation maps supervised by ground-truths. At inference (b), we use top-k strategy to predict existing categories and reject non-existing categories for next decoder layer.

3.3. Category Early Rejection

The computational cost of gradual fusion decoder is proportional to the number of categories. When the number of categories is very large, the inference time significantly increases. In fact, most images only contain several categories. As a result, majority of inference time of the decoder is taken to calculate the features of non-existing categories. To boost inference speed, we introduce a category early rejection scheme to recognize these existing categories and reject non-existing categories at early decoder layer. The feature maps corresponding to rejected categories are removed from current decoder layer, and the following decoder layer only considers the reserved categories.

During training, as shown in Fig. 4(a), we add the auxiliary convolutional branch after each layer to respectively predict segmentation maps, which are supervised by ground-truths. To avoid the negative effect on model training, we stop their gradient back-propagation to the decoder.

During inference, we employ a top-k strategy on segmentation maps to predict the existing semantic categories. Specifically, we select the top-k categories with maximum responses for each pixel and generate a union set of categories from all pixels, which is fed to next decoder layer. We observe that $k = 8$ can ensure that most existing categories is recognized. Fig. 4(b) shows the category early rejection during inference. We first predict segmentation maps from $\mathcal{F}^{l1}_{dec}$ and employ the top-k strategy to select N_{l1} categories. Then, we remove the feature maps of non-selected categories and generate the output feature map $\mathcal{F}^{l1}_{cer} \in \mathbb{R}^{H_v \times W_v \times N_{l1} \times D}$. The generated feature map $\mathcal{F}^{l1}_{cer}$ is fed to the decoder layer. Similarly, we generate the feature maps with fewer categories for the following layers. Therefore, most non-existing categories are rejected at early layer, which boosts the inference speed of decoder.

4. Experiments

4.1. Datasets and Evaluation Metric

Following existing open-vocabulary semantic segmentation methods [12, 56], we use the large-scale dataset COCO-Stuff [4] to train the model. In COCO-Stuff dataset, the training set contains about $118k$ densely-annotated images with 171 different semantic categories. With the trained model on COCO-Stuff, we conduct experiments on multiple widely-used semantic segmentation datasets (ADE20K [61], PASCAL VOC [17], and PASCAL-Context [34]) to demonstrate the effectiveness of proposed SED and compare it with state-of-the-art methods in literature.

ADE20K [61] is a large-scale semantic segmentation dataset. It contains $20k$ training images and $2k$ validation images. In open-vocabulary semantic segmentation task, there are two different test sets: `A-150` and `A-847`. The test set `A-150` has 150 common categories, while the test set `A-847` has 847 categories.

PASCAL VOC [17] is one of the earliest datasets for object detection and segmentation. There are about $1.5k$ training images and $1.5k$ validation images. The dataset contains 20 different object categories. In open-vocabulary semantic segmentation task, we name it as `PAS-20`.

PASCAL-Context [34] is extended from the original PASCAL VOC dataset for semantic segmentation. In open-vocabulary semantic segmentation, there are two different test sets: `PC-59` and `PC-459`. The test set `PC-59` has 59 categories, while the test set `PC-459` has 459 categories.

Evaluation metric: Following existing traditional and open-vocabulary semantic segmentation, we adopt mean Intersection over Union (mIoU) as evaluation metric. It is the averaged value of intersection over unions over all classes.

4.2. Implementation Details

We adopt the pre-trained vision-language model CLIP [11, 38, 41] as the base model, where the hierarchical backbone ConvNeXt-B or ConvNeXt-L is used as hierarchical image encoder. The feature dimension D_t of text embeddings are 640 for ConvNeXt-B and 768 for ConvNeXt-L, the number of category templates P is 80, and the channel number of feature map F^{l1}_{dec} is 128. We freeze the text encoder following most open-vocabulary methods, and only train the image encoder and gradual fusion decoder. We train our model on 4 NVIDIA A6000 GPUs with the mini-batch of 4 images. The optimizer AdamW is adopted with the initial learning rate of 2×10^{-4} and the weight decay of 1×10^{-4}. To avoid over-fitting on training set, the learning rate of image encoder is multiplied by a factor of $\lambda = 0.01$. There are totally $80k$ training iterations. During training, we crop the input image with the 768×768 pixels. During inference, the input image is resized with the 768×768 pixels.

Method	VLM	Feature backbone	Training dataset	A-847	PC-459	A-150	PC-59	PAS-20
SPNet [51]	-	ResNet-101	PASCAL VOC	-	-	-	24.3	18.3
ZS3Net [3]	-	ResNet-101	PASCAL VOC	-	-	-	19.4	38.3
LSeg [28]	ViT-B/32	ResNet-101	PASCAL VOC-15	-	-	-	-	47.4
LSeg+ [19]	ALIGN	ResNet-101	COCO-Stuff	2.5	5.2	13.0	36.0	-
Han et al. [23]	ViT-B/16	ResNet-101	COCO Panoptic [27]	3.5	7.1	18.8	45.2	83.2
GroupViT [53]	ViT-S/16	-	GCC [42]+YFCC [46]	4.3	4.9	10.6	25.9	50.7
ZegFormer [14]	ViT-B/16	ResNet-101	COCO-Stuff-156	4.9	9.1	16.9	42.8	86.2
ZegFormer [12]	ViT-B/16	ResNet-101	COCO-Stuff	5.6	10.4	18.0	45.5	89.5
SimBaseline [55]	ViT-B/16	ResNet-101	COCO-Stuff	7.0	-	20.5	47.7	88.4
OpenSeg [19]	ALIGN	ResNet-101	COCO Panoptic [27]+LOc. Narr. [37]	4.4	7.9	17.5	40.1	-
DeOP [22]	ViT-B/16	ResNet-101c	COCO-Stuff-156	7.1	9.4	22.9	48.8	91.7
PACL [35]	ViT-B/16	-	GCC [42]+YFCC [46]	-	-	31.4	50.1	72.3
OVSeg [29]	ViT-B/16	ResNet-101c	COCO-Stuff+COCO Caption	7.1	11.0	24.8	53.3	92.6
CAT-Seg [12]	ViT-B/16	ResNet-101	COCO-Stuff	8.4	16.6	27.2	**57.5**	93.7
SAN [56]	ViT-B/16	-	COCO-Stuff	10.1	12.6	27.5	53.8	94.0
SED (Ours)	ConvNeXt-B	-	COCO-Stuff	**11.4**	**18.6**	**31.6**	57.3	**94.4**
LSeg [28]	ViT-B/32	ViT-L/16	PASCAL VOC-15	-	-	-	-	52.3
OpenSeg [19]	ALIGN	Eff-B7 [45]	COCO Panoptic [27]+LOc. Narr. [37]	8.1	11.5	26.4	44.8	-
OVSeg [29]	ViT-L/14	Swin-B	COCO-Stuff+COCO Caption	9.0	12.4	29.6	55.7	94.5
Ding *et al.* [15]	ViT-L/14	-	COCO Panoptic [27]	8.2	10.0	23.7	45.9	-
ODISE [54]	ViT-L/14	-	COCO Panoptic [27]	11.1	14.5	29.9	57.3	-
HIPIE [47]	BERT-B [13]	ViT-H	COCO Panoptic [27]	-	-	29.0	59.3	-
SAN [56]	ViT-L/14	-	COCO-Stuff	13.7	17.1	33.3	60.2	95.5
CAT-Seg [12]	ViT-L/14	Swin-B	COCO-Stuff	10.8	20.4	31.5	**62.0**	**96.6**
FC-CLIP [58]	ConvNeXt-L	-	COCO Panoptic [27]	**14.8**	18.2	34.1	58.4	95.4
SED (Ours)	ConvNeXt-L	-	COCO-Stuff	13.9	**22.6**	**35.2**	60.6	96.1

Table 1. **Comparison with state-of-the-art methods.** We report the mIoU results on five widely used test sets for open-vocabulary semantic segmentation. Here, the best results are shown in bold and the second-best results are underlined. With comparable VLM model, our proposed SED achieves superior performance on all five test sets.

Method	mIoU	Time
SimBaseline [55]	20.5	316
OVSeg [29]	24.8	314
CAT-Seg [12]	27.2	362
SAN [56]	27.5	**32**
SED (ours)	**31.6**	82
SED-fast (ours)	29.4	**32**

(a) With base model

Method	mIoU	Time
ODISE [54]	29.9	1989
CAT-Seg [12]	31.5	433
SAN [56]	33.3	117
FC-CLIP [58]	34.1	285
SED (ours)	**35.2**	98
SED-fast (ours)	34.2	**64**

(b) With large model

Table 2. **Comparison in terms of mIoU and inference time (ms).** We report the results on A-150 with base and large models. Here, the inference time is reported on a single NVIDIA A6000 GPU.

HECG	GFD	CER	A-847	PC-459	A-150	PC-59	PAS-20
			7.3	14.9	23.7	52.9	94.4
✓			9.9	17.2	28.2	54.7	95.0
✓	✓		11.2	18.6	31.8	57.7	94.4
✓	✓	✓	11.4	18.6	31.6	57.3	94.4

Table 3. **Impact of different modules in our SED.** We show the results of integrating different modules into the baseline.

4.3. Comparisons With State-of-the-art Methods

Table 1 compares our SED with state-of-the-art open-vocabulary semantic segmentation methods on five different test sets. Most existing methods are developed on VLM with plain transformer ViT, including two-stage OVSeg [29] and single-stage CAT-Seg [12] and SAN [56]. In contrast, our proposed SED adopts hierarchical encoder ConvNeXt. When using the comparable image encoder ViT-B or ConvNeXt-B, our SED outperforms these methods on all five test sets. On PC-459, our SED outperforms OVSeg [29], CAT-Seg [12], and SAN [56] by 7.6%, 2.0%, and 6.0%. On A-150, our SED outperforms OpenSeg [19], CAT-Seg [12], and SAN [56] by 14.1%, 4.4%, and 4.1%. Moreover, compared to OVSeg and CAT-Seg, our SED does not require additional feature backbone. Compared to OVSeg and OpenSeg, our SED does not require additional dataset or annotation.

When using the comparable image encoder ViT-L or ConvNeXt-L, our SED also achieves favourable performance on all five test sets. For example, on PC-459, our SED outperforms SAN [56], CAT-Seg [12], and FC-CLIP [58] by 5.5%, 2.2%, and 4.4%.

Table 2 further shows the accuracy and speed comparison on A-150. Compared to most methods, our proposed SED has the results with both base and large models at fast speed. We also present a faster version (SED-fast) by downsampling the input size. Compared to SAN, our SED-fast is 1.9% better at similar speed with base model, and is 0.9% better and 1.8 times faster with large model.

4.4. Ablation Study

Here we perform ablation study to show the efficacy of our proposed method using ConvNeXt-B as image encoder.

Impact of integrating different components: Table 3 shows the impact of integrating different components into

Image Encoder	Skip-layer	A-847	PC-459	A-150	PC-59	PAS-20
Plain	w/o	7.3	13.5	23.0	51.5	94.1
	with	7.3	14.9	23.7	52.9	94.4
Hierarchical	w/o	7.9	14.3	25.7	52.0	92.7
	with	9.9	17.2	28.2	54.7	95.0

Table 4. **Comparison of plain and hierarchical encoder.** We employ ViT-B and ConvNeXt-B as plain encoder and hierarchical encoder, respectively.

	Strategy	A-847	PC-459	A-150	PC-59	PAS-20
(a)	Freeze All	9.4	15.3	28.6	49.7	77.2
	Freeze L0-L2	11.2	18.3	30.6	54.9	91.5
	Freeze L0-L1	10.4	17.5	31.4	57.3	94.0
	Freeze L0	10.6	17.7	31.6	57.2	94.1
	Fine-tune All	11.2	18.6	31.8	57.7	94.4
	Factor λ	A-847	PC-459	A-150	PC-59	PAS-20
(b)	0.005	11.3	17.6	31.6	56.4	93.6
	0.01	11.2	18.6	31.8	57.7	94.4
	0.02	10.5	17.7	31.3	57.7	94.7

Table 5. **Ablation study on fine-tuning image encoder in HECG.** We show the results of different fine-tuning strategies and different scale factors of encoder learning rates.

the baseline. The baseline adopts original CLIP with plain transformer ViT and uses the cost map to predict segmentation map with skip-layer fusion. The baseline obtains the mIoU scores of 7.3%, 14.9%, and 23.7% on A-847, PC-459, and A-150. When using hierarchical encoder to replace plain transformer, it has the mIoU scores of 9.9%, 17.2%, and 28.2% on A-847, PC-459, and A-150, outperforming the baseline by 2.6%, 2.3%, and 4.5%. When further integrating gradual fusion decoder, it has the mIoU scores of 11.2%, 18.6%, and 31.8% on A-847, PC-459, and A-150, outperforming the baseline by 3.9%, 3.7%, and 8.1%. When further integrating category early rejection strategy into our method, it almost does not degrade performance but has a faster speed (see Table 7).

Plain *vs* hierarchical encoder: Table 4 compares plain ViT-B or hierarchical ConvNeXt-B used as image encoder. Hierarchical encoder outperforms plain encoder with or without using skip-layer connection. When using skip-layer connection, hierarchical encoder has a larger improvement. Therefore, different feature maps of hierarchical encoder provide rich local information for segmentation.

Ablation study on fine-tuning encoder in HECG: Table 5 presents ablation study on fine-tuning hierarchical encoder in HECG. In top part (a), we show the impact of different fine-tuning strategies. When we freeze all the layers in the encoder, it has the lowest performance on all five test sets. When we fine-tune all the layers in the encoder, it achieves the best results on all test sets. In bottom part (b), we present the impact of scale factor λ of encoder learning rates. When the scale factor is 1×10^{-2}, it has the best performance. A larger or smaller scale factor will degrade the performance in some degree. Therefore, we fine-tune hierarchical encoder using the scale factor of 1×10^{-2}.

	Kernel Size	A-847	PC-459	A-150	PC-59	PAS-20
(a)	7	11.1	18.0	31.8	57.3	93.9
	9	11.2	18.6	31.8	57.7	94.4
	11	10.8	18.0	31.8	57.1	94.5
	Aggregation	A-847	PC-459	A-150	PC-59	PAS-20
(b)	Spatial-level	10.1	17.6	28.8	54.8	94.4
	Class-level	10.0	17.4	30.7	56.0	92.7
	Both	11.2	18.6	31.8	57.7	94.4
	Feature Fusion	A-847	PC-459	A-150	PC-59	PAS-20
(c)	None	7.9	14.3	25.7	52.0	92.7
	+$\mathcal{F}_{2,3,4}$	11.1	17.9	32.0	57.5	94.2
	+$\mathcal{F}_{2,3,4}$+$\mathcal{F}_{cv}$	11.2	18.6	31.8	57.7	94.4
	Gradient to $\mathcal{F}_{2,3,4}$	A-847	PC-459	A-150	PC-59	PAS-20
(d)	w/o Stop	10.6	18.0	31.7	57.3	93.9
	with Stop	11.2	18.6	31.8	57.7	94.4
	Decoder Layer	A-847	PC-459	A-150	PC-59	PAS-20
(e)	1	10.1	17.0	29.8	55.6	91.0
	2	11.1	18.3	31.8	57.3	93.8
	3	11.2	18.6	31.8	57.7	94.4

Table 6. **Ablation study on different designs in GFD.** Gradually fusion decoder (GFD) contains feature aggregation module (FAM) and skip-layer fusion module (SFM). We first show the impact of different kernel sizes (a) and spatial-class aggregation (b) in FAM. Then, we give the impact of fusing different feature maps (c) and gradient back-propagation (d) in SFM. Finally, we show the impact of different decoder layers (e).

Ablation study on GFD: Table 6 presents ablation study on different designs in gradual fusion decoder (GFD). Our gradual fusion decoder contains feature aggregation module (FAM) and skip-layer fusion module (SFM). We first perform some experiments on FAM. In (a), we show the impact of different large-kernel sizes. It has the best results using the kernel size of 9. We also observe that large-kernel operation is better and faster than Swin block. On PC-459, the mIoU has 0.4% improvement and the speed is 1.2 times faster. In (b), we present the impact of spatial-level and class-level feature aggregation. Both spatial-level and class-level feature aggregation improve performance on five test sets. When combining them together, it has the best results.

Afterwards, we perform some experiments on SFM. In (c), we present the impact of fusing different feature maps in SFM. It can be seen that integrating different feature maps can significantly improve performance. In (d), we present the impact of gradient back-propagation in SFM. It has better performance without gradient back-propagation. Finally, we show the impact of different decoder layers in (e). Compared to using only one decoder layer, using three decoder layers has the best results. For example, it has 1.6% improvement on PC-459.

Ablation study on CER: Table 7 presents the impact of selecting top-k categories in category early rejection (CER). The small number of k indicates selecting few categories to fed the decoder layer, which can accelerate inference speed but may sacrifice accuracy. For example, on A-847, when k is equal to 1, the speed is 7.1 times faster than using all

Figure 5. **Qualitative results.** In the left part, we show some high-quality results, where our method can accurately classify and segment various categories. In the right-top part, we give some failure cases and corresponding ground-truths (GT). In the right-bottom part, we give one case in which our method can segment the cat that is not annotated in ground-truths (GT).

Metric	k	A-847	PC-459	A-150	PC-59	PAS-20
mIoU	1	10.2	17.6	29.9	55.9	93.8
	2	10.9	18.3	30.8	56.7	94.0
	4	11.2	18.5	31.4	57.0	94.2
	8	11.4	18.6	31.6	57.3	94.4
	All	11.2	18.6	31.8	57.7	94.4
Time (ms)	1	120.8	74.8	51.3	40.5	37.2
	2	136.4	84.2	55.8	44.2	38.9
	4	151.5	93.4	67.5	49.7	42.2
	8	181.6	120.1	82.4	59.8	47.3
	All	861.0	468.1	177.7	84.7	44.9

Table 7. **Impact of selecting top-k categories in CER.** We show both mIoU and inference time (ms). Here, the inference time is reported on a single NVIDIA A6000 GPU, and k=all represents keeping all categories in each dataset (namely not using CER).

categories, but the mIoU is 1.0% lower. When $k = 8$, we observe a slight improvement in performance, with a speed increase of about 4.7 times.

Impact of transforming SED to CAT-Seg: We presents the impact of gradually replacing our encoder-decoder with CAT-Seg encoder-decoder in the Appendix, demonstrating the efficacy and efficiency of our encoder-decoder.

Qualitative results: Fig. 5 presents some qualitative results. The left part shows high-quality segmentation results. Our method is able to accurately segment various categories, such as palm, runway, and sand. The right-top part shows some failure cases. In first two rows, our method mistakenly recognizes the water as sea, the earth as rock, and the car as truck. In third row, our method mistakenly recognizes airplane and ignores windowpane. In addition, the right-bottom part shows that our method successfully segments cat that is ignored by the PC-59 ground-truth.

5. Conclusion

We propose an approach, named SED, for open-vocabulary semantic segmentation. Our SED comprises hierarchical encoder-based cost map generation and gradual fusion decoder with category early rejection. We first employ hierarchical encoder to generate pixel-level image-text cost map. Based on generated cost map and different feature maps in hierarchical encoder, we employ gradual fusion decoder to generate high-resolution feature map for segmentation. To boost speed, we introduce a category early rejection scheme into decoder to early reject non-existing categories. Experiments on multiple datasets reveal the effectiveness of our method in terms of both accuracy and speed.

Future work: Our model sometimes struggle on recognizing near-synonym categories as classes. In future, we will explore designing category attention strategy or using large-scale fine-grained dataset to solve this challenge.

Acknowledgement: This work was supported by the National Key Research and Development Program of China (No. 2022ZD0160400), Natural Science Foundation of China (No. 62271346, 62206031).

References

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

[2] Yifan Bai, Zeyang Zhao, Yihong Gong, and Xing Wei. Artrackv2: Prompting autoregressive tracker where to look and how to describe. *arXiv preprint arXiv:2312.17133*, 2023.

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

[4] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1209–1218, 2018.

[5] Jiale Cao, Yanwei Pang, and Xuelong Li. Triply supervised decoder networks for joint detection and segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7384–7393, 2019.

[6] 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.

[7] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European Conference on Computer Vision*, pages 104–120. Springer, 2020.

[8] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*, 2022.

[9] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *Advances in Neural Information Processing Systems*, pages 17864–17875, 2021.

[10] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022.

[11] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023.

[12] Seokju Cho, Heeseong Shin, Sunghwan Hong, Seungjun An, Seungjun Lee, Anurag Arnab, Paul Hongsuck Seo, and Seungryong Kim. Cat-seg: Cost aggregation for open-vocabulary semantic segmentation. *arXiv preprint arXiv:2303.11797*, 2023.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[14] Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11583–11592, 2022.

[15] Zheng Ding, Jieke Wang, and Zhuowen Tu. Open-vocabulary panoptic segmentation with maskclip. *arXiv preprint arXiv:2208.08984*, 2022.

[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, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88:303–338, 2010.

[18] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019.

[19] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *European Conference on Computer Vision*, pages 540–557, 2022.

[20] Jiaqi Gu, Hyoukjun Kwon, Dilin Wang, Wei Ye, Meng Li, Yu-Hsin Chen, Liangzhen Lai, Vikas Chandra, and David Z Pan. Multi-scale high-resolution vision transformer for semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12094–12103, 2022.

[21] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *International Conference on Learning Representations*, pages 1–20, 2022.

[22] Cong Han, Yujie Zhong, Dengjie Li, Kai Han, and Lin Ma. Zero-shot semantic segmentation with decoupled one-pass network. In *IEEE/CVF International Conference on Computer Vision*, pages 1086–1096, 2023.

[23] Kunyang Han, Yong Liu, Jun Hao Liew, Henghui Ding, Jiajun Liu, Yitong Wang, Yansong Tang, Yujiu Yang, Jiashi Feng, Yao Zhao, et al. Global knowledge calibration for fast open-vocabulary segmentation. In *IEEE/CVF International Conference on Computer Vision*, pages 797–807, 2023.

[24] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *IEEE/CVF International Conference on Computer Vision*, pages 603–612, 2019.

[25] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916, 2021.

[26] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165, 2020.

[27] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019.

[28] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations*, pages 1–13, 2022.

[29] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7061–7070, 2023.

[30] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022.

[31] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[32] Chenyang Lu, Daan de Geus, and Gijs Dubbelman. Content-aware token sharing for efficient semantic segmentation with vision transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23631–23640, 2023.

[33] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, pages 13–23, 2019.

[34] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 891–898, 2014.

[35] Jishnu Mukhoti, Tsung-Yu Lin, Omid Poursaeed, Rui Wang, Ashish Shah, Philip HS Torr, and Ser-Nam Lim. Open vocabulary semantic segmentation with patch aligned contrastive learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19413–19423, 2023.

[36] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters–improve semantic segmentation by global convolutional network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4353–4361, 2017.

[37] Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. Connecting vision and language with localized narratives. In *European Conference on Computer Vision*, pages 647–664, 2020.

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763, 2021.

[39] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6148–6157, 2017.

[40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer Assisted Intervention*, pages 234–241, 2015.

[41] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In *Advances in Neural Information Processing Systems*, pages 25278–25294, 2022.

[42] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Association for Computational Linguistics*, pages 2556–2565, 2018.

[43] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *IEEE/CVF International Conference on Computer Vision*, pages 7262–7272, 2021.

[44] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019.

[45] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114, 2019.

[46] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Association for Computing Machinery*, 59(2):64–73, 2016.

[47] Xudong Wang, Shufan Li, Konstantinos Kallidromitis, Yusuke Kato, Kazuki Kozuka, and Trevor Darrell. Hierarchical open-vocabulary universal image segmentation. *arXiv preprint arXiv:2307.00764*, 2023.

[48] Xing Wei, Yifan Bai, Yongchao Zheng, Dahu Shi, and Yihong Gong. Autoregressive visual tracking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9697–9706, 2023.

[49] Dongming Wu, Wencheng Han, Tiancai Wang, Xingping Dong, Xiangyu Zhang, and Jianbing Shen. Referring multi-object tracking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14633–14642, 2023.

[50] Dongming Wu, Wencheng Han, Tiancai Wang, Yingfei Liu, Xiangyu Zhang, and Jianbing Shen. Language prompt for autonomous driving. *arXiv preprint arXiv:2309.04379*, 2023.

[51] Yongqin Xian, Subhabrata Choudhury, Yang He, Bernt Schiele, and Zeynep Akata. Semantic projection network for zero-and few-label semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8256–8265, 2019.

[52] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Advances in Neural Information Processing Systems*, pages 12077–12090, 2021.

[53] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18134–18144, 2022.

[54] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2955–2966, 2023.

[55] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In *European Conference on Computer Vision*, pages 736–753, 2022.

[56] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. Side adapter network for open-vocabulary semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2945–2954, 2023.

[57] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. Denseaspp for semantic segmentation in street scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3684–3692, 2018.

[58] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. *arXiv preprint arXiv:2308.02487*, 2023.

[59] Hang Zhao, Xavier Puig, Bolei Zhou, Sanja Fidler, and Antonio Torralba. Open vocabulary scene parsing. In *IEEE/CVF International Conference on Computer Vision*, pages 2002–2010, 2017.

[60] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6881–6890, 2021.

[61] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127:302–321, 2019.

[62] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *European Conference on Computer Vision*, page 696–712, 2022.