SegCLIP: Patch Aggregation with Learnable Centers for Open-Vocabulary Semantic Segmentation

Huaishao Luo [1 2] **Junwei Bao** [1] **Youzheng Wu** [1] **Xiaodong He** [1] **Tianrui Li** [2]

Abstract

Recently, the contrastive language-image pre-training, e.g., CLIP, has demonstrated promising results on various downstream tasks. The pre-trained model can capture enriched visual concepts for images by learning from a large scale of text-image data. However, transferring the learned visual knowledge to open-vocabulary semantic segmentation is still under-explored. In this paper, we propose a CLIP-based model named SegCLIP for the topic of open-vocabulary segmentation in an annotation-free manner. The SegCLIP achieves segmentation based on ViT and the main idea is to gather patches with learnable centers to semantic regions through training on text-image pairs. The gathering operation can dynamically capture the semantic groups, which can be used to generate the final segmentation results. We further propose a reconstruction loss on masked patches and a superpixel-based KL loss with pseudo-labels to enhance the visual representation. Experimental results show that our model achieves comparable or superior segmentation accuracy on the PASCAL VOC 2012 (+0.3% mIoU), PASCAL Context (+2.3% mIoU), and COCO (+2.2% mIoU) compared with baselines. We release the code at https://github.com/ArrowLuo/SegCLIP.

1. Introduction

Semantic segmentation, aiming to assign a label to each pixel of a given image, is an important task and has been researched for a long time. The CNN-based approaches (Long et al., 2015; Ronneberger et al., 2015; Chen et al., 2015; Zhao et al., 2017; Chen et al., 2018; Wen et al., 2022) and Transformer-based approaches (Cheng et al., 2021; Zheng et al., 2021; Xie et al., 2021; Cheng et al., 2022; Jain et al., 2022) have achieved impressive performance on this topic. However, two significant limitations still need exploration: expensive pixel-level labeling and restricted labeled categories leading to weak generalization (Bucher et al., 2019; Xian et al., 2019).

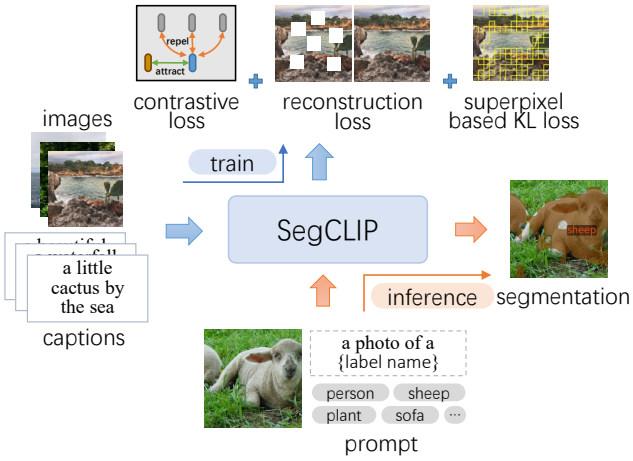

Figure 1. **Overview of our problem**. The proposed SegCLIP can achieve open-vocabulary semantic segmentation through training with image-text pairs.

Recent works propose to leverage large-scale image-text pre-trained models to alleviate the above limitations. These works involve zero-shot or weakly supervised semantic segmentation because the large image-text pairs are class-agnostic. Due to the target being to segment an image with arbitrary categories instead of fixed labeling vocabularies, this kind of method is also called open-vocabulary semantic segmentation (Ghiasi et al., 2021; Xu et al., 2022b; Liang et al., 2022; Ma et al., 2022). They can be roughly divided into two types. The first is the classification-based method, supervised by the extracted pseudo labels or text features from a pre-trained model, e.g., CLIP (Radford et al., 2021). Moreover, this type is usually achieved with a fully convolutional network or carries out prediction based on mask

[1]JD AI Research [2]Southwest Jiaotong University, Chengdu, China. Correspondence to: Huaishao Luo <huaishaoluo@gmail.com>, Junwei Bao <baojunwei001@gmail.com>, Tianrui Li <trli@swjtu.edu.cn>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

proposals (Zhou et al., 2022a; Xu et al., 2022b). The other is to group semantic regions along training with large-scale image-text datasets, which can be called the group-based method (Xu et al., 2022a). Through different routes, the fundamental logic behind them is that the image-text pre-trained model can learn vision-text alignment from image-level to pixel-level features. Some interpretability methods, like CAM (Selvaraju et al., 2017) and Transformer-interpretability (Chefer et al., 2021), can support such an argument, such as in the work of (Zabari & Hoshen, 2021).

Following the research line of learning pixel-level alignment from image-text pairs, we explore the semantic regions with the group-based method in this paper. Compared with the classification-based method, which involves mask proposals and label classification, the group-based method is straightforward. It has consistent objectives with the pretraining model, e.g., training with a contrastive loss using image-text pairs. Further, the group-based model jointly learns visual and textual representations as humans do, so it has the potential to be improved from a multimodal perspective. Instead of training from scratch, the group-based method can also benefit from the pre-trained model.

To this end, we propose a group-based model SegCLIP to accomplish open-vocabulary semantic segmentation. The SegCLIP can be regarded as *segmentation*+CLIP. Specifically, the proposed model has a similar architecture to the CLIP but a modified image encoder. The image encoder is based on the ViT (Vision Transformer) (Dosovitskiy et al., 2021). Instead of operating on regular grids, we designed a plugged semantic group module to aggregate patches with learnable centers. The learnable centers can dynamically merge visual patches to semantic concepts via a mapping matrix generated by a cross-attention mechanism. This plugged group module can be inserted into the middle layers of the image encoder to generate irregular-shaped segments. Thus, the SegCLIP can transfer knowledge from CLIP to semantic segmentation. We use a small number of image-text pairs to train our experiments' extra randomly initialized parameters. Figure 1 illustrates the training and inference process. During inference, the label name is filled into a given prompt format, and the semantic segments are obtained by calculating the similarity between the text representation and the semantic groups.

Moreover, we propose two auxiliary losses to enhance the visual representation for semantic segmentation. One is a reconstruction loss, which aims to recover the masked patches through their visual context. Such a reconstruction loss is effective from the previous work (He et al., 2022; Wang et al., 2022a; Zhou et al., 2022b). The difference is that our reconstruction process is designed based on irregular-shaped segments with a mapping matrix instead of regular patches. The other is a KL loss (Kullback-Leibler divergence Loss) used to learn a better mapping matrix via the superpixel label, which can be obtained via the off-the-shelf tool. The KL loss can keep the consistency of pixel-level features.

2. Model

Figure 2 presents the SegCLIP as a dual-encoder architecture. One encoder is for text representation, and the other is for image representation. We propose a plugged semantic group module to aggregate patches with learnable centers in the image encoder, thus injecting the CLIP with the capacity to deal with semantic segmentation. The backbone of SegCLIP is the ViT version of CLIP, and the details can be found in (Radford et al., 2021). We describe the architecture of SegCLIP, training losses, and inference process in detail in this section.

2.1. Main Architecture

The architecture of SegCLIP mainly consists of a text encoder $\mathcal{E}_T(\cdot)$ and an image encoder $\mathcal{E}_I(\cdot)$, similar to the CLIP. Such a design can transfer the knowledge naturally from the pre-trained weights of the CLIP instead of training from scratch. Nevertheless, it takes work to achieve semantic segmentation directly because the CLIP is pre-trained with image-level features and needs help to finish pixel-level tasks. We propose a plugged semantic group module within the image encoder with learnable centers to aggregate the low-layer pixel features to achieve the segmentation. The learnable centers can be regarded as semantic regions and gather semantical pixels along with the training process. Thus, the SegCLIP can finish open-vocabulary semantic segmentation.

As shown in Figure 2, the model's input is a pair of text $\mathbf{T} = \{w_i\}_{i=1}^M$ and image $\mathbf{I} = \{p_j\}_{j=1}^N$, where w_i means the i-th token within the text, p_j means the j-th non-overlapped patches of the image, M and N denotes the number of given text and image, respectively. Following the ViT version of CLIP, the token is generated via a lower-cased byte pair encoding (BPE), and the tokens representation $\{\mathbf{e}_{w_i}\}_{i=1}^M$ and patches representation $\{\mathbf{e}_{p_j}\}_{j=1}^N$ are obtained by an Embedding operation and a linear projection, respectively. Then the tokens representation is fed into Transformer layers (Vaswani et al., 2017) to generate the final text feature as $\mathbf{z}_w = \mathcal{E}_T\big(\{\mathbf{e}_{w_i}\}_{i=1}^M\big)$. The image representation is fed into other Transformer layers plus the semantic group module to generate the final image feature as $\mathbf{z}_p = \mathcal{E}_I\big(\{\mathbf{e}_{p_j}\}_{j=1}^N\big)$. Finally, the contrastive loss can be calculated on the text feature $\mathbf{z}_w$ and the image feature $\mathbf{z}_p$. In our setting, the text feature $\mathbf{z}_w$ comes from a special token [SEP], which is appended as the last token of the text. The image feature $\mathbf{z}_p$ is generated by the last Transformer layer followed by a `max-pooling` operation.

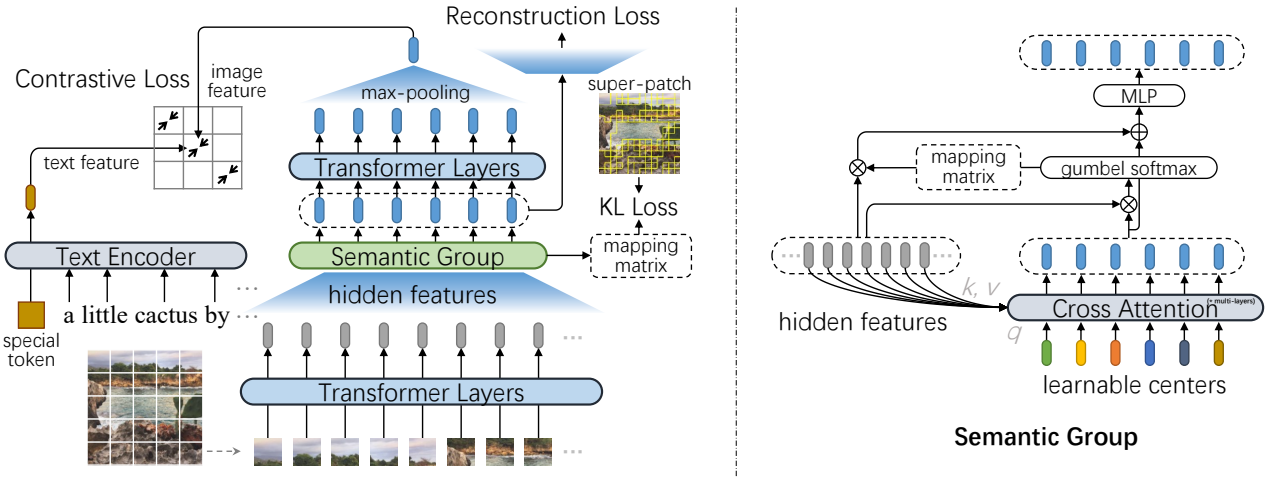

Figure 2. **The framework of SegCLIP**. The SegCLIP is a dual-encoder architecture containing a text and image encoder. The semantic group module (zoom in at the right) is proposed to generate regular patches to arbitrary-shaped semantic regions. Three losses, including contrastive loss, reconstruction loss, and superpixel-based KL loss, are used in training.

2.2. Semantic Group Module

To gather the regular patches to arbitrary-shaped semantic regions, we design a semantic group to plug into the Transformer layers of the image encoder. In other words, the semantic group module can be regarded as the second stage of the image encoder, with different Transformer layers as the first and third stages. Assuming the patches representation is $\mathcal{H}_p = \{\mathbf{h}_{p_j}^s\}_{j=1}^N$ after passing through the first stage's s-th (also the last) Transformer layer. The semantic group module can gather different patches by calculating semantic similarity. Specifically, we first randomly initialize a group of learnable centers $\mathcal{H}_c = \{\mathbf{c}_k\}_{k=1}^L$, then obtain contextual centers $\hat{\mathcal{H}}_c = \{\hat{\mathbf{c}}_k\}_{k=1}^L$ through some cross-attention layers as follows,

$$\hat{\mathcal{H}}_c^t = \texttt{CrossAttention}(\mathcal{H}_c^t, \mathcal{H}_p, \mathcal{H}_p), \tag{1}$$

where t is the layer number of cross-attention, the start $\mathcal{H}_c^1$ is the $\mathcal{H}_c$, and the $\hat{\mathcal{H}}_c$ is the last $\hat{\mathcal{H}}_c^t$, the `CrossAttention` is a cross-attention layer, the same as the Self-Attention layer in Transformer (Vaswani et al., 2017), but the input is asymmetrically separate embedding sequences, in here, the query is $\mathcal{H}_c^t$, and the key and value are $\mathcal{H}_p$, respectively.

After obtaining the contextual centers $\hat{\mathcal{H}}_c$, we can assign each patch to a corresponding center via a mapping matrix $\mathcal{M} \in \mathbb{R}^{N \times L}$ generated by the `Gumbel-Softmax` operation (Jang et al., 2017; Xu et al., 2022a).

$$\mathcal{M} = \texttt{Gumbel-Softmax}(\mathcal{H}_p \hat{\mathcal{H}}_c^\top), \tag{2}$$

where each row of $\mathcal{M}$ is a one-hot vector, and $\mathcal{M}_{jk}$ denotes the j-th patch belongs to k-th semantic center if its value is 1. The $\mathcal{M}$ keeps a patch belonging to only and if only a center, which benefits the final semantic segmentation.

Finally, we can calculate the representation of semantic regions $\hat{\mathcal{H}}_p$ with the patches representation $\mathcal{H}_p$, the mapping matrix $\mathcal{M}$, and the contextual centers $\hat{\mathcal{H}}_c$ as follows,

$$\hat{\mathcal{H}}_p = \texttt{MLP}\big(\texttt{MEAN}(\mathcal{M}^\top \mathcal{H}_p) + \hat{\mathcal{H}}_c\big), \tag{3}$$

where `MEAN` denotes doing the average for each center using the patches belonging to it. `MLP` is a multilayer perceptron block containing two fully-connected layers and a GELU (Hendrycks & Gimpel, 2016) between them.

The generated representation of semantic regions $\hat{\mathcal{H}}_p$ is fed to the Transformer layers of the third stage to learn sufficiently interactive region features $\mathcal{Z}_p$ further.

2.3. Reconstruction Loss

In addition to the contrastive loss, we propose a self-supervised reconstruction loss to enhance the visual representation for segmentation. As shown in Figure 3, the reconstruction loss aims to recover the masked patches through their visual context, similar to MAE (He et al., 2022). The difference is that our reconstruction process is designed based on irregular-shaped segments with a mapping matrix.

We first generate a masked version of region representation $\hat{\mathcal{H}}_p^{(m)}$ and mapping matrix $\mathcal{M}^{(m)}$ via the semantic group module on the unmasked patches for the MAE encoder. However, the region representation can not be used to calculate the reconstruction loss because the unmasked patches have been gathered into different regions. We propose a reconstruction layer to restore the representation of patches from $\hat{\mathcal{H}}_p^{(m)}$ as,

$$\tilde{\mathcal{H}}_p^{(m)} = \text{GELU}\big(\texttt{Linear}(\mathcal{M}^{(m)})^\top \hat{\mathcal{H}}_p^{(m)}\big), \tag{4}$$

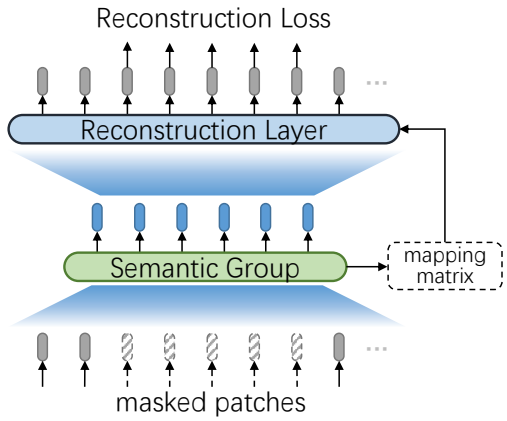

Figure 3. **Reconstruction Loss**.

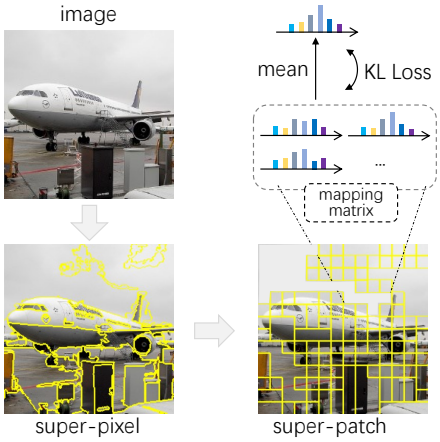

Figure 4. **Superpixel based KL Loss**.

where `Linear` is a fully-connected layer, and the GELU is the activation function. Then we use extra Transformer layers, similar to the third stage of the image encoder, to obtain the final representation $\mathcal{Z}_p^{(m)}$ using the $\tilde{\mathcal{H}}_p^{(m)}$.

We keep the MAE decoder as in (He et al., 2022) with the input $\mathcal{Z}_p^{(m)}$. Finally, the reconstruction loss is the mean squared error (MSE) between the reconstructed image $\mathbf{I}^{(m)}$ and the original image $\mathbf{I}$, $\mathcal{L}_{rec} = \texttt{MSE}(\mathbf{I}^{(m)}, \mathbf{I})$.

2.4. Superpixel based KL Loss

Besides the reconstruction loss, we propose a superpixel-based KL loss to guild the learning of a mapping matrix. The motivation is to keep the pixel-level consistency when gathering the patches to regions. Intuitively, the pixels of a superpixel should be gathered into a region instead of one more region. The calculation process is illustrated in Figure 4. For a given image, we first obtain its superpixel with a graph-based segmentation method from (Felzenszwalb & Huttenlocher, 2004), which is unsupervised and does not need to train on any datasets. There are many other superpixel methods, but we chose this typical one as a demonstration.

Assuming there are some superpixels, each pixel in the same superpixels has the same label, e.g., superpixel id. Thus for each patch, we assign it a label, e.g., the average floor value of ids from its pixels. Thus, we can obtain a super-patch corresponding to the superpixel. Intuitively, the patches within a super-patch are also covered by a superpixel. Note that a superpixel id is a number used to distinguish different superpixels, and we do not care about its meaning in the loss calculation. Every patch of a super-patch should have a consistent probability in the mapping matrix $\mathcal{M}$ because they should be gathered in a region. In other words, the probability of a patch in the mapping matrix should be similar to the average probability of the patches within the same super-patch. Thus, a symmetric KL loss is designed as follows,

$$\hat{\mathcal{P}}_j = \texttt{softmax}\Big(\frac{1}{|\mathcal{G}_j|}\sum_{\hat{j}\in\mathcal{G}_j}\mathcal{P}_{\hat{j}}\Big), \tag{5}$$

$$\mathcal{L}_{sup} = \frac{1}{2N}\sum_{j=1}^{N}\big(\texttt{KL}(\mathcal{P}_j,\hat{\mathcal{P}}_j)+\texttt{KL}(\hat{\mathcal{P}}_j,\mathcal{P}_j)\big), \tag{6}$$

where $\texttt{KL}$ is the Kullback-Leibler divergence, $\mathcal{P}_j$ is the regions' probability of j-th patch, which is obtained by the j-th row of $\mathcal{M}$ after $\texttt{softmax}$ operation, and $\mathcal{G}_j$ is the indexes of the patches contained in a super-patch which also contains the j-th patch. By decreasing the $\mathcal{L}_{sup}$, the model tends to gather the patches within a superpixel together, which benefits the segmentation.

2.5. Training and Inference

Training In addition to the reconstruction loss $\mathcal{L}_{rec}$ and the superpixel-based KL loss $\mathcal{L}_{sup}$, the model is also trained with the contrastive loss $\mathcal{L}_{con}$ in an end-to-end manner. The total loss is the sum of them,

$$\mathcal{L}_{total} = \mathcal{L}_{con} + \mathcal{L}_{rec} + \mathcal{L}_{sup}. \tag{7}$$

The $\mathcal{L}_{con}$ is a symmetric cross-entropy loss calculated on the text feature $\mathbf{z}_w$ and image feature $\mathbf{z}_p$, similar to the CLIP (Radford et al., 2021),

$$\mathcal{L}_{con} = \frac{1}{2}(\mathcal{L}_{t2i} + \mathcal{L}_{i2t}), \tag{8}$$

$$\mathcal{L}_{t2i} = -\frac{1}{\mathcal{B}}\sum_{i}^{\mathcal{B}}\log\frac{\exp\big(s(\mathbf{z}_w^{(i)}, \mathbf{z}_p^{(i)})\big)}{\sum_{j=1}^{\mathcal{B}}\exp\big(s(\mathbf{z}_w^{(j)}, \mathbf{z}_p^{(i)})\big)}, \tag{9}$$

$$\mathcal{L}_{i2t} = -\frac{1}{\mathcal{B}}\sum_{i}^{\mathcal{B}}\log\frac{\exp\big(s(\mathbf{z}_w^{(i)}, \mathbf{z}_p^{(i)})\big)}{\sum_{j=1}^{\mathcal{B}}\exp\big(s(\mathbf{z}_w^{(i)}, \mathbf{z}_p^{(j)})\big)}, \tag{10}$$

where, $s(\mathbf{z}_i, \mathbf{z}_j) = \frac{\mathbf{z}_i \mathbf{z}_j^\top}{\|\mathbf{z}_i\| \|\mathbf{z}_j\|}$ is the cosine similarity, $\mathcal{B}$ is the batch size, the superscripts (i) and (j) means the i-th and j-th sample, respectively.

Inference Due to the learned mapping matrix, the SegCLIP can finish semantic segmentation without further finetuning on any datasets. The image feature can be obtained through the image encoder for a segmentation task with candidate labels. For the text feature, we use a template `a photo of a {label name}.` to form the input text of the text encoder with different label names. Specifically, we use the $\mathcal{Z}_p \in \mathbb{R}^{L \times H}$ from the last Transformer layer of the image encoder as the representation of each region, where L is the number of learnable centers, and H is the hidden size. The label features can be denoted as $\{\mathbf{z}_w^{(\tau)}\}_{\tau=1}^{\mathcal{T}}$ if there are $\mathcal{T}$ candidate labels. After calculating the cosine similarity of each row of $\mathcal{Z}_p$ and each $\mathbf{z}_w^{(\tau)}$ via $s(\mathbf{z}_i, \mathbf{z}_j)$ from Eqs. (9-10), we can obtain a similarity matrix $\hat{\mathcal{S}} \in \mathbb{R}^{L \times \mathcal{T}}$, in which each row denotes the labels' probability of a region. Then the similarity $\mathcal{S} \in \mathbb{R}^{N \times \mathcal{T}}$ between patches and candidate labels can be calculated using the mapping matrix $\mathcal{M} \in \mathbb{R}^{N \times L}$ via $\mathcal{S} = \mathcal{M}\hat{\mathcal{S}}$. We can assign each patch a label with the highest similarity of each row from the $\mathcal{S}$. We can further execute an interpolation operation on the $\mathcal{S}$ from N to image size to obtain a pixel-level assignment matrix, then an irregular-shaped and pixel-level segmentation.

3. Experiments

We first describe datasets and implementation details before ablating various settings of our model. Then we present the state-of-the-art results on three datasets in an annotation-free manner. Finally, we demonstrate some qualitative results of our model.

3.1. Datasets

We pretrain the SegCLIP on the training splits of Conceptual Captions (CC) (Sharma et al., 2018) and COCO (Lin et al., 2014), which contain 3M and 400K image-text pairs, respectively.

For the semantic segmentation, we evaluate the model on the validation splits of the PASCAL VOC 2012 (Everingham et al., 2010), PASCAL Context (Mottaghi et al., 2014), and COCO datasets. These datasets contain 20, 59, and 80 foreground classes, respectively. To distinguish the foreground classes from the background, we set the threshold to 0.75, 0.25, and 0.65 on the similarities for PASCAL VOC 2012, PASCAL Context, and COCO, respectively. The metric is the mIoU calculated with the predicted and ground truth segmentation masks. The short side of the given image is resized to 224 during inference.

3.2. Experimental Details

The architecture is based on the ViT version of CLIP, and the text encoder and image encoder are all 12 Transformer layers. The image size is set to 224 $\times$ 224, and the patch size is 16 $\times$ 16. The max length of the text tokens is 32. We initialize the embedding and Transformer layers from the CLIP pre-trained weight as default. For the semantic group module, we put it after the 10th Transformer layer in the image encoder via grid search based on segmentation datasets. The cross-attention layer number is set to 2. The decoder layer of MAE is 3, and the mask rate of patches is 0.75. The number of learnable centers is 8. We randomly initialize the parameters of the semantic group module, MAE decoder, and the rest of the Linears in the model. For the optimization, we use Adam optimizer and a cosine schedule of learning rate following the CLIP. The initial learning rate is 4e-6 for the embedding layers, text encoder, and Transformer layers of the image encoder before the semantic group module. For the rest of the parameters, the initial learning rate is 4e-3. We pretrain our model using 8 NVIDIA A100 GPUs with a batch size of 768 for 10 epochs. This process takes approximately 6 hours.

3.3. Ablation Studies

We conduct ablation studies on the designed losses and key hyperparameters to introduce their influence in this section.

Effective of the Reconstruction Loss As shown in Table 1, training with the reconstruction loss can improve the mIoU by 1.19%, 0.92%, and 0.66% on PASCAL VOC, PASCAL Context, and COCO, respectively under the condition of without the superpixel-based KL loss and can improve the mIoU by 4.11%, 0.56%, and 0.52% under the condition of with the superpixel based KL loss. The results demonstrate that the well-designed reconstruction loss restoring the masked patches with contextual visual features can enhance the encoder and make the mapping matrix a better match for the segmentation task.

Effective of the Superpixel based KL Loss We also report the consistent improvement of the superpixel-based KL loss in Table 1. The gains are 0.54%, 0.72%, and 1.07%, and 3.46%, 0.36%, and 0.93% on PASCAL VOC, PASCAL Context, and COCO under the condition of without or with the reconstructing loss, respectively. We suppose that the pseudo superpixel labels can keep the pixel-level visual feature relatively consistent within the segments.

Influence of the Plugged Layer In Table 2, we conduct experiments on different plugged layers of the semantic group module, from 6 to 11, with the same 8 learnable centers. The results reflect that the plugged layer 10 can achieve better performance than other plugged points, and

Model	R-Loss	S-KL	VOC	Context	COCO
SegCLIP			47.95	23.43	24.86
SegCLIP	✓		49.14	24.35	25.52
SegCLIP		✓	48.49	24.15	25.93
SegCLIP	✓	✓	**52.60**	**24.71**	**26.45**

Table 1. **Ablation of the proposed losses (mIoU)**. *R-Loss* is the reconstruction loss, *S-KL* is the superpixel-based KL loss.

Model	P-Ly	C-NO.	VOC	Context	COCO
SegCLIP	6	8	35.28	19.28	16.73
SegCLIP	8	8	43.75	22.71	21.40
SegCLIP	10	6	47.03	23.36	24.85
SegCLIP	10	8	**47.95**	23.43	**24.86**
SegCLIP	10	10	44.89	**23.46**	24.74
SegCLIP	11	8	22.07	10.76	12.08

Table 2. **Ablation of plugged layer (P-Ly) and center number (C-NO.) of semantic group module**. The results are obtained with only the contrastive loss.

too small and big numbers decrease the mIoU significantly. We suppose that too small plugged points may harm the pre-trained CLIP weights, and the low-layer feature is segments-irrelevant. The features from big plugged points are also segments-irrelevant and can not benefit the segmentation task.

Influence of the Center Number We also conduct experiments on different learnable centers of the semantic group module with the same plugged layer 10. The results in Table 2 demonstrate that the 8 learnable centers can achieve better or comparable performance. The mIoU is not sensitive on 6, 8, and 10 learnable centers. We chose 8 as the default hyperparameter in this work.

Influence of the Cross-Attention Layer Table 3 shows the results on different layers of the cross-attention layer in the semantic group module. Compared with the mIoU obtained by training without a cross-attention layer, training with a cross-attention layer can achieve better performance. Such a phenomenon suggests that the cross-attention layer can make the learnable centers match better with the features of patches and focus on different parts of the given image. We also obverse that two cross-attention layers achieve better mIoU than others, but the results from numbers 1 and 3 are comparable. A large number of cross-attention layers, e.g., 4, may harm the performance. We consider that our training datasets are insufficient to train deep layers.

Model	Cross-Att.	VOC	Context	COCO
SegCLIP	0	44.44	22.28	22.11
SegCLIP	1	47.63	23.29	24.21
SegCLIP	2	**47.95**	**23.43**	**24.86**
SegCLIP	3	47.83	23.24	24.70
SegCLIP	4	45.39	23.17	23.80

Table 3. **Ablation of cross-attention layer (Cross-Att.)** The plugged layer is 10, and the center NO. is 8. The results are obtained with only the contrastive loss.

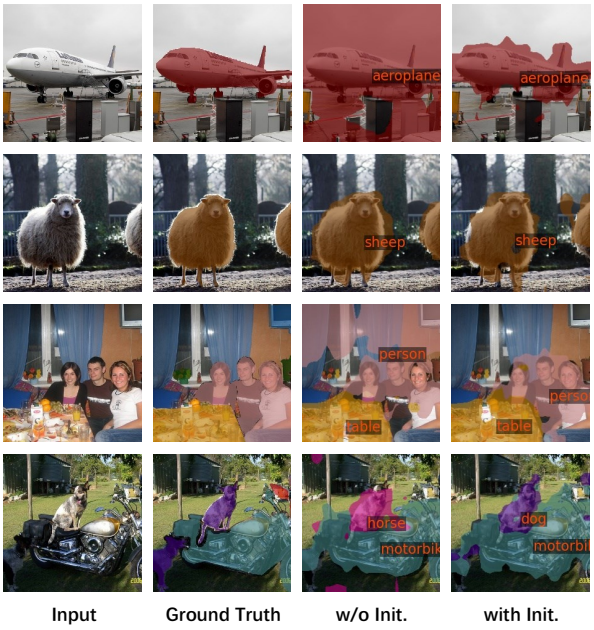

Figure 5. **Qualitative results on PASCAL VOC**.

3.4. Comparisons with State-of-the-Art Methods

As shown in Table 4, we compare our model against class-supervised, visually self-supervised, and textually supervised baselines. The results of class-supervised and visually self-supervised baselines are obtained from (Xu et al., 2022a). They are pixel-wise classification models finetuned on the pre-trained ViT models, i.e., DeiT (Touvron et al., 2021), DINO (Caron et al., 2021), and MoCo (Chen et al., 2021), with a 1×1 convolutional layer as the semantic segmentation head. The finetuning datasets are the training sets of the VOC and Context separately. Compared with the class-supervised model, our result (52.5%) on VOC is still comparable (53.0%), although training without manually pixel-level annotations.

Compared with the state-of-the-art textually supervised method GroupViT, our initialized SegCLIP achieves 0.3%, 2.3%, and 2.2% gains on the VOC, Context, and COCO, respectively. We also conduct experiments for GroupViT on

Model	Arch.	Init.	Training Data	Sup.	Zero-Shot	VOC	Context	COCO
DeiT♮ (Touvron et al., 2021)	ViT		ImageNet	Class	✗	53.0	35.9	-
DINO♮ (Caron et al., 2021)	ViT		CC12M+YFCC	Self	✗	37.6	22.8	-
MoCo♮ (Chen et al., 2021)	ViT		CC12M+YFCC	Self	✗	36.1	23.0	-
GroupViT (Xu et al., 2022a)	ViT		CC12M+YFCC	Text	✓	52.3	22.4	24.3
GroupViT$_{1\text{-s}}$	ViT		CC+COCO	Text	✓	28.1	14.8	12.9
GroupViT$_{2\text{-s}}$	ViT		CC+COCO	Text	✓	19.7	10.4	8.0
SegCLIP (ours)	ViT		CC+COCO	Text	✓	33.3	19.1	15.2
SegCLIP (ours)	ViT	✓	CC+COCO	Text	✓	**52.6**	**24.7**	**26.5**

Table 4. **Comparison of different models on mIoU**. 'Arch.' and 'Sup.' are short for architecture and supervision, respectively. 'Init.' means whether be initialized with CLIP. CC12M and YFCC are from (Changpinyo et al., 2021) and (Thomee et al., 2016), respectively. ♮ means results from (Xu et al., 2022a). GroupViT$_{1\text{-s}}$ and GroupViT$_{2\text{-s}}$ are our implementations on the CC and COCO datasets, with one-stage and two-stage grouping blocks, respectively.

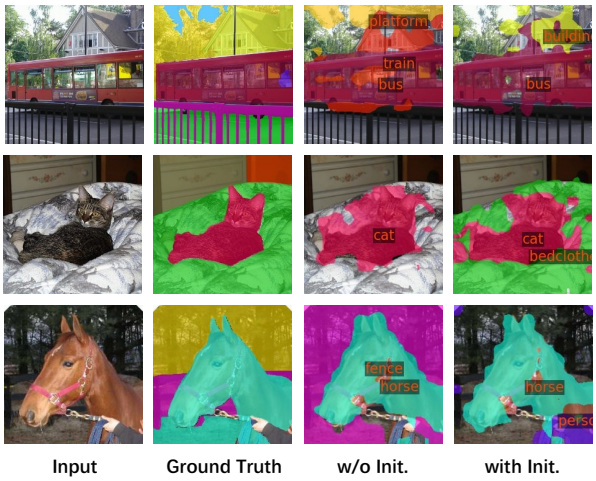

Figure 6. **Qualitative results on PASCAL Context**.

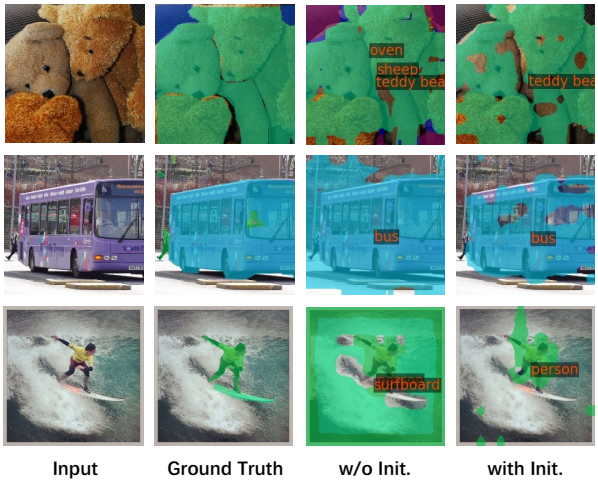

Figure 7. **Qualitative results on COCO**.

CC and COCO datasets for a fair comparison. Our SegCLIP trained from scratch achieves 5.2%, 4.3%, and 2.3% improvements compared with the GroupViT$_{1\text{-s}}$. Note that the GroupViT$_{1\text{-s}}$ achieves superior accuracy than GroupViT$_{2\text{-s}}$ in our settings. We suppose the CC and COCO, which are smaller than the CC12M (Changpinyo et al., 2021) and YFCC (Thomee et al., 2016), may lead the unstable and insufficient training for the 2-stage GroupViT. When initialized with the pre-trained CLIP, the SegCLIP improves the mIoU by 19.3%, 5.6%, and 11.3% on the VOC, Context, and COCO compared with training from scratch, respectively. It implies that our model could benefit from the pre-trained CLIP, which also proves the flexibility of the semantic ground module.

3.5. Qualitative Results

We demonstrate the qualitative results on PASCAL VOC, PASCAL Context, and COCO in Figures 5-7, respectively. The results indicate that the SegCLIP can generate plausible segments and reasonable tags. Compared with the SegCLIP training from scratch, the initialized SegCLIP achieves better semantic segmentation. In Figure 5, the first row implies the initialized SegCLIP could obtain better semantics, e.g., the airplane area, and the last row presents the initialized SegCLIP could obtain correct tags, e.g., dog, for the generated segments. The same conclusion could be drawn from Figure 6 and the first two rows of Figure 7. We can also observe that the single objective, multiple objects of the same class, or multiple objects from different classes can be captured by the SegCLIP. It suggests that the model training on the large scale of image-text pairs could induce the fine-grained alignment between segments and tags.

4. Related Work

This paper is related to the vision-language pre-training and open-vocabulary semantic segmentation.

4.1. Vision-Language pre-training

The vision-language pre-training (VLP) is an emerging research topic with the increase of large-scale visual and linguistic pairs collected from the Internet (Tan & Bansal, 2019; Chen et al., 2020; Huang et al., 2020; Kim et al., 2021; Li et al., 2021; Wang et al., 2022b; Sun et al., 2019; Luo et al., 2020; Bain et al., 2021; Li et al., 2022b;c). The research directions commonly involve the design of new model architectures and pre-training objectives (Gan et al., 2022). For the architecture, the VLP models usually contain several modules, e.g., visual encoder, text encoder, multi-modal fusion encoder, or decoder. For the objective, the representative pre-training tasks contain the masked language model (MLM) introduced in language pre-training (Devlin et al., 2019), vision-text matching (VTM), vision-text contrastive learning (VTC), and masked vision model (MVM). Besides the model, the available datasets are the key factor in pushing the development of this research field, e.g., Conceptual Captions (Sharma et al., 2018) and COCO (Lin et al., 2014) used in this work, CC12M (Changpinyo et al., 2021), YFCC (Thomee et al., 2016), LAION-400M (Schuhmann et al., 2021), and HowTo100M (Miech et al., 2019).

Most vision-language pre-training models are designed for image-text or video-text related downstream tasks. Beyond that, some are mainly designed for visual tasks with text supervision. The CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021) are two typical models trained with VTC for image classification. Further, the works from (Yao et al., 2022; Zeng et al., 2022) consider fine-grained alignment, and the work from (Li et al., 2022e) considers self-supervision within each modality plus other tasks when pre-training the model. There are also some pretrain models for object detection (Gu et al., 2022; Zhong et al., 2022; Li et al., 2022d) and segmentation (Wu et al., 2020; Ghiasi et al., 2021; Lüddecke & Ecker, 2022; Rao et al., 2022; Ding et al., 2022b).

The proposed SegCLIP is a vision-language pre-training model for segmentation. Besides our elaborate model, we also designed a reconstruct loss and a superpixel-based KL loss as our training objectives. Although the SegCLIP is capable of training with a large-scale dataset, we consider its transfer capability of reusing the existing pre-trained model, i.e., CLIP, for segmentation and reducing the cost of training resources.

4.2. Open-Vocabulary Semantic Segmentation

The open-vocabulary semantic segmentation also called semantic segmentation in the wild in the literature, has been widely researched along with vision-text pretraining. Its target is to segment an image with arbitrary categories described by texts instead of fixed labeling vocabularies. As a pioneering work, ZS3Net (Bucher et al., 2019) combines a deep visual segmentation model with a generative model of class-dependent features. Such architecture allows the generation of visual samples from unseen classes via training a classifier with real visual samples from seen classes. SPNet (Xian et al., 2019) achieves that by transferring the knowledge from previously seen classes to novel classes by incorporating class-level semantic information into any network designed for semantic segmentation.

Due to the impressive zero-shot transferability of CLIP (Radford et al., 2021) on various downstream tasks, a research line is to leverage it for open-vocabulary semantic segmentation. DenseCLIP (Rao et al., 2022) is a dense prediction framework that converts the original image-text matching problem in CLIP to a pixel-text matching problem and uses the pixel-text score maps to guide the learning of dense prediction models. Unlike DenseCLIP, which needs an image decoder to generate the segments and is trained with ground-truth labels, MaskCLIP (Zhou et al., 2022a) uses pseudo per-pixel labels generated from CLIP and self-training to achieve annotation-free segmentation. Similarly, (Zabari & Hoshen, 2021) uses model interpretability to obtain pixel-level pseudo-labels from CLIP to supervise single-image segmentation methods. ZegFormer (Ding et al., 2022a) decouples the zero-shot semantic segmentation into two sub-tasks, i.e., grouping the pixels into segments and classifying the segments with the CLIP. CLIPSeg (Lüddecke & Ecker, 2022) is a system building upon the CLIP model as a backbone and can generate image segmentations based on arbitrary prompts. OpenSeg (Ghiasi et al., 2021) also involves proposal generation and segments classification as the ZegFormer, but it needs training with class agnostic mask annotations to generate mask proposals.

Similarly, ZSSeg (Xu et al., 2022b) proposes a two-stage semantic segmentation framework, with the first stage generating mask proposals and the second stage leveraging CLIP to classify the generated proposals. LSeg (Li et al., 2022a) uses a text encoder to provide a flexible label representation with a transformer-based image encoder trained with a contrastive objective to align pixel embeddings to the text embedding of the corresponding semantic class. OVSeg (Liang et al., 2022) proposes to finetune CLIP on a collection of masked image regions and their corresponding text descriptions. Fusioner (Ma et al., 2022) is a simple, lightweight cross-modality fusion module that can be used to explicitly bridge a variety of self-supervised pre-trained visual/language models for open-vocabulary semantic segmentation.

Unlike previous works, our model does not require any mask proposals or segmentation decoders. Instead, the proposed model uses a plugged semantic group module to aggregate patches as segments. Our work follows the line of GroupViT (Xu et al., 2022a), which learns segmentation masks from

text supervision. However, we have different architecture compared with the GroupViT, and the proposed semantic group module makes the model capable of resuing the pre-trained weights from CLIP and training from scratch with noisy image-text pairs. Moreover, we propose two novel objectives to improve the visual representation further.

5. Conclusion and Future Work

This paper proposes a CLIP-based model SegCLIP for weakly-supervised semantic segmentation. The model could generate plausible segmentation results with only training on the annotation-free text-image datasets. The process does not contain the training on the labels or even the seen classes of segmentation datasets before inference, demonstrating a solid transfer character. The other advantage is the flexibility of the plugged design of the semantic group module, which brings the possibility of reusing the pre-trained CLIP weights. Besides, the proposed reconstruction loss and the superpixel-based KL loss improve performance, indicating that the image encoder's encoding capacity is essential for the semantics before giving the proper tags. In summary, the work takes a further step toward achieving fine-grained alignment, e.g., semantic segmentation in this paper, from training only on a large scale of image-text pairs.

Limitations The SegCLIP utilizes an interpolation operation to smooth the predicted boundaries. However, it was found that the use of regular image patches as the input to the image encoder often leads to rough predictions. The study recommends reducing the patch size to achieve smoother and more precise boundaries.

Future Work To demonstrate the advantages of using smaller patch sizes, we conduct experiments on the VOC, Context, and COCO datasets using a patch size of 32, resulting in 49 patches per image. The obtained mIoU scores are 44.2%, 22.0%, and 21.4% for each dataset, respectively. By comparing these scores to the mIoU scores of 52.5%, 24.7%, and 26.5% achieved with a patch size of 16, which equates to 196 patches per image, it becomes evident that larger patch sizes can lead to inferior performance. Therefore, future research could concentrate on pretraining models with smaller patch sizes.

Furthermore, additional experiments are performed using the validation sets from ADE20K (Zhou et al., 2017) and Cityscapes (Cordts et al., 2016) to assess more complex scenes. The results show that SegCLIP achieves mIoU scores of 8.7% and 11.0% on ADE20K and Cityscapes, respectively, meanwhile GroupViT$_{1\text{-}s}$ achieves 4.9% and 4.2% mIoU on the same datasets. It is revealed that the complexity and intricacy of the scenes play a crucial role in performance, suggesting that exploring complex scenes is a promising research direction in the field of open-vocabulary segmentation.

The current superpixel generation process in SegCLIP operates as an offline module and is not trained end-to-end, making it essential to explore innovative end-to-end training techniques for this module to enhance its effectiveness. Additionally, the use of finely-divided superpixels may result in biased patches, making it necessary to consider the use of class-agnostic segmentation methods to generate improved pseudo-labels and improve performance. Furthermore, although the SegCLIP framework can utilize the pre-trained CLIP model as initialization, there is still potential for further advancements through post-pretraining on more extensive datasets such as CC12M and YFCC. These future research directions will present both challenges and exciting opportunities to enhance SegCLIP's performance to new heights.

Acknowledgments

This work was supported by the National Key R&D Program of China (No. 2020AAA0108600) and the National Science Foundation of China (No. 62176221).

References

Bain, M., Nagrani, A., Varol, G., and Zisserman, A. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, pp. 1708–1718, 2021.

Bucher, M., Vu, T., Cord, M., and Pérez, P. Zero-shot semantic segmentation. In *NeurIPS*, 2019.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *ICCV*, pp. 9630–9640, 2021.

Changpinyo, S., Sharma, P., Ding, N., and Soricut, R. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, pp. 3558–3568, 2021.

Chefer, H., Gur, S., and Wolf, L. Transformer interpretability beyond attention visualization. In *CVPR*, pp. 782–791, 2021.

Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

Chen, X., Xie, S., and He, K. An empirical study of training self-supervised vision transformers. In *ICCV*, pp. 9620–9629, 2021.

Chen, Y., Li, L., Yu, L., Kholy, A. E., Ahmed, F., Gan, Z., Cheng, Y., and Liu, J. UNITER: universal image-text representation learning. In *ECCV*, volume 12375, pp. 104–120, 2020.

Cheng, B., Schwing, A. G., and Kirillov, A. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, pp. 17864–17875, 2021.

Cheng, B., Misra, I., Schwing, A. G., Kirillov, A., and Girdhar, R. Masked-attention mask transformer for universal image segmentation. In *CVPR*, pp. 1290–1299, 2022.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pp. 3213–3223, 2016.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pp. 4171–4186, 2019.

Ding, J., Xue, N., Xia, G., and Dai, D. Decoupling zero-shot semantic segmentation. In *CVPR*, pp. 11573–11582, 2022a.

Ding, Z., Wang, J., and Tu, Z. Open-vocabulary panoptic segmentation with maskclip. *arXiv preprint arXiv:2208.08984*, 2022b.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

Everingham, M., Gool, L. V., Williams, C. K. I., Winn, J. M., and Zisserman, A. The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vis.*, 88(2):303–338, 2010.

Felzenszwalb, P. F. and Huttenlocher, D. P. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004.

Gan, Z., Li, L., Li, C., Wang, L., Liu, Z., and Gao, J. Vision-language pre-training: Basics, recent advances, and future trends. *arXiv preprint arXiv:2210.09263*, 2022.

Ghiasi, G., Gu, X., Cui, Y., and Lin, T.-Y. Scaling open-vocabulary image segmentation with image-level labels. *arXiv:2112.12143*, 2021.

Gu, X., Lin, T.-Y., Kuo, W., and Cui, Y. Open-vocabulary object detection via vision and language knowledge distillation. In *ICLR*, 2022.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. B. Masked autoencoders are scalable vision learners. In *CVPR*, pp. 15979–15988, 2022.

Hendrycks, D. and Gimpel, K. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Huang, Z., Zeng, Z., Liu, B., Fu, D., and Fu, J. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *arXiv preprint arXiv:2004.00849*, 2020.

Jain, J., Li, J., Chiu, M., Hassani, A., Orlov, N., and Shi, H. Oneformer: One transformer to rule universal image segmentation. *arXiv preprint arXiv:abs/2211.06220*, 2022.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2017.

Jia, C., Yang, Y., Xia, Y., Chen, Y., Parekh, Z., Pham, H., Le, Q. V., Sung, Y., Li, Z., and Duerig, T. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, volume 139, pp. 4904–4916, 2021.

Kim, W., Son, B., and Kim, I. Vilt: Vision-and-language transformer without convolution or region supervision. In *ICML*, volume 139, pp. 5583–5594, 2021.

Li, B., Weinberger, K. Q., Belongie, S., Koltun, V., and Ranftl, R. Language-driven semantic segmentation. In *ICLR*, 2022a.

Li, J., Li, D., Xiong, C., and Hoi, S. C. H. BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, volume 162, pp. 12888–12900, 2022b.

Li, L., Gan, Z., Lin, K., Lin, C., Liu, Z., Liu, C., and Wang, L. LAVENDER: unifying video-language understanding as masked language modeling. *arXiv preprint arXiv:2206.07160*, 2022c.

Li, L. H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J., Chang, K., and Gao, J. Grounded language-image pre-training. In *CVPR*, pp. 10955–10965, 2022d.

Li, W., Gao, C., Niu, G., Xiao, X., Liu, H., Liu, J., Wu, H., and Wang, H. UNIMO: towards unified-modal understanding and generation via cross-modal contrastive learning. In *ACL/IJCNLP*, pp. 2592–2607, 2021.

Li, Y., Liang, F., Zhao, L., Cui, Y., Ouyang, W., Shao, J., Yu, F., and Yan, J. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. In *ICLR*, 2022e.

Liang, F., Wu, B., Dai, X., Li, K., Zhao, Y., Zhang, H., Zhang, P., Vajda, P., and Marculescu, D. Open-vocabulary semantic segmentation with mask-adapted CLIP. *arXiv preprint arXiv:abs/2210.04150*, 2022.

Lin, T., Maire, M., Belongie, S. J., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft COCO: common objects in context. In *ECCV*, volume 8693, pp. 740–755, 2014.

Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. In *CVPR*, pp. 3431–3440, 2015.

Lüddecke, T. and Ecker, A. S. Image segmentation using text and image prompts. In *CVPR*, pp. 7076–7086, 2022.

Luo, H., Ji, L., Shi, B., Huang, H., Duan, N., Li, T., Chen, X., and Zhou, M. Univl: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*, 2020.

Ma, C., Yang, Y., Wang, Y., Zhang, Y., and Xie, W. Open-vocabulary semantic segmentation with frozen vision-language models. *arXiv preprint arXiv:abs/2210.15138*, 2022.

Miech, A., Zhukov, D., Alayrac, J., Tapaswi, M., Laptev, I., and Sivic, J. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, pp. 2630–2640, 2019.

Mottaghi, R., Chen, X., Liu, X., Cho, N., Lee, S., Fidler, S., Urtasun, R., and Yuille, A. L. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, pp. 891–898, 2014.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In *ICML*, volume 139, pp. 8748–8763, 2021.

Rao, Y., Zhao, W., Chen, G., Tang, Y., Zhu, Z., Huang, G., Zhou, J., and Lu, J. Denseclip: Language-guided dense prediction with context-aware prompting. In *CVPR*, pp. 18061–18070, 2022.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pp. 234–241, 2015.

Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., and Komatsuzaki, A. LAION-400M: open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, pp. 618–626, 2017.

Sharma, P., Ding, N., Goodman, S., and Soricut, R. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, pp. 2556–2565, 2018.

Sun, C., Myers, A., Vondrick, C., Murphy, K., and Schmid, C. Videobert: A joint model for video and language representation learning. In *ICCV*, pp. 7463–7472, 2019.

Tan, H. and Bansal, M. LXMERT: learning cross-modality encoder representations from transformers. In *EMNLP*, 2019.

Thomee, B., Shamma, D. A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., and Li, L. YFCC100M: the new data in multimedia research. *Commun. ACM*, 59(2): 64–73, 2016.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. In *ICML*, volume 139, pp. 10347–10357, 2021.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *NeurIPS*, pp. 5998–6008, 2017.

Wang, R., Chen, D., Wu, Z., Chen, Y., Dai, X., Liu, M., Jiang, Y.-G., Zhou, L., and Yuan, L. Bevt: Bert pretraining of video transformers. In *CVPR*, pp. 14733–14743, 2022a.

Wang, Z., Yu, J., Yu, A. W., Dai, Z., Tsvetkov, Y., and Cao, Y. Simvlm: Simple visual language model pretraining with weak supervision. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022b.

Wen, X., Zhao, B., Zheng, A., Zhang, X., and Qi, X. Self-supervised visual representation learning with semantic grouping. In *NeurIPS*, 2022.

Wu, C., Lin, Z., Cohen, S., Bui, T., and Maji, S. Phrasecut: Language-based image segmentation in the wild. In *CVPR*, pp. 10213–10222, 2020.

Xian, Y., Choudhury, S., He, Y., Schiele, B., and Akata, Z. Semantic projection network for zero- and few-label semantic segmentation. In *CVPR*, pp. 8256–8265, 2019.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., and Luo, P. Segformer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, pp. 12077–12090, 2021.

Xu, J., De Mello, S., Liu, S., Byeon, W., Breuel, T., Kautz, J., and Wang, X. Groupvit: Semantic segmentation emerges from text supervision. In *CVPR*, pp. 18134–18144, 2022a.

Xu, M., Zhang, Z., Wei, F., Lin, Y., Cao, Y., Hu, H., and Bai, X. A simple baseline for open vocabulary semantic segmentation with pre-trained vision-language model. *ECCV*, 2022b.

Yao, L., Huang, R., Hou, L., Lu, G., Niu, M., Xu, H., Liang, X., Li, Z., Jiang, X., and Xu, C. FILIP: fine-grained interactive language-image pre-training. In *ICLR*, 2022.

Zabari, N. and Hoshen, Y. Semantic segmentation in-the-wild without seeing any segmentation examples. *arXiv:2112.03185*, 2021.

Zeng, Y., Zhang, X., and Li, H. Multi-grained vision language pre-training: Aligning texts with visual concepts. In *ICML*, volume 162, pp. 25994–26009, 2022.

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. Pyramid scene parsing network. In *CVPR*, pp. 6230–6239, 2017.

Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P. H., and Zhang, L. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, 2021.

Zhong, Y., Yang, J., Zhang, P., Li, C., Codella, N., Li, L. H., Zhou, L., Dai, X., Yuan, L., Li, Y., and Gao, J. Regionclip: Region-based language-image pretraining. In *CVPR*, pp. 16772–16782, 2022.

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., and Torralba, A. Scene parsing through ade20k dataset. In *CVPR*, pp. 633–641, 2017.

Zhou, C., Loy, C. C., and Dai, B. Extract free dense labels from clip. In *ECCV*, 2022a.

Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A. L., and Kong, T. iBOT: Image BERT pre-training with online tokenizer. In *ICLR*, 2022b.