A Survey on Open-Vocabulary Semantic Segmentation: Challenges, Methods and Future

Guanlin Liu
School of Computer Science and Engineering, Sun Yat-sen University
liuglin8@mail2.sysu.edu.cn

Abstract

*Semantic segmentation is a critical task in computer vision, aiming to classify each pixel in an image into a specific category at a pixel-level resolution.Due to the high cost of manual labeling, the annotated categories in existing datasets are typically small in scale and predefined. This means that models perform image segmentation within a fixed set of predefined categories and cannot generalize beyond this closed vocabulary, a limitation known as Closed Vocabulary Segmentation(**CVS**).To address this limitation, the community has been growing interest in Open Vocabulary Segmentation (**OVS**)) over the past few years. OVS is an extension of semantic segmentation that aims to segment and identify new categories not encountered during training. It can recognize and segment both known and novel categories, using natural language descriptions to guide segmentation.This review presents a comprehensive overview of the latest advancements in OVS. It summarizes the state-of-the-art OVS methods and recent research on models, discussing their research motivations, technical challenges, and major contributions. In conclusion, this review presents and discusses several promising research directions, aiming to inspire and guide future studies in the field.*

1. Introduction

Semantic segmentation is a core problem in computer vision, with the aim of partitioning an image into coherent regions with their respective semantic class labels. Over the past decade, models based on Convolutional Neural Networks (CNNs) and Transformers [6, 18, 21], have achieved substantial and consistent advancements in segmentation tasks. To date, they have been widely applied in various fields, such as autonomous driving [2] and medical image analysis [21]. However, traditional semantic segmentation methods operate on a closed vocabulary, meaning they can only recognize and segment a fixed number of classes that were present during training. Given the diversity and dynamism of the real world, it is impractical to have labeled data for every possible object or category a model might encounter.

To overcome the constraint imposed by a closed vocabulary in scene perception tasks, **Open Vocabulary Semantic Segmentation (OVSS)** has attracted significant attention as a burgeoning research field. It aims to surpass the constraints of traditional semantic segmentation by enabling models to recognize and segment categories that were not seen during training. Unlike traditional methods, OVSS is not limited to a fixed set of predefined categories but leverages the generalization and transfer learning abilities of models to identify and segment new categories.

The realization of OVSS depends on the integration of deep learning and natural language processing techniques [16]. Specifically, the use of large-scale pre-trained vision-language models, such as CLIP [20], has proven to be an effective strategy. These models are jointly trained on extensive image and text datasets, learning to map images and texts into a shared embedding space, thereby enabling open vocabulary recognition. However, effectively utilizing these pre-trained models and adapting and optimizing them for specific tasks remains a significant challenge.

This review aims to systematically summarize and analyze the latest developments in OVSS. First, we will introduce the basic concepts of OVSS in detail. In the second section, we will review the background of the OVSS task, including traditional semantic segmentation methods, and the history and roadmap of OVSS. Then, we will introduce the preliminaries related to OVSS, covering methods based on vision-language models, cross-domain transfer learning methods, and self-supervised learning methods, as well as summarizing the commonly used evaluation metrics and benchmark datasets. In the third section, we will categorize OVSS methods into four types based on their model architecture: Pixel-Level Feature Matching, Two-Stage Region-Level Image-Text Matching, Improved Single-Stage, and Open Segmentation. We will introduce representative papers from these four categories progressively, highlighting

the research motivations and key technical points. Finally, we will summarize the existing challenges and suggest future research directions.

2. Background

2.1. Problem Definition

The task of OVSS is to perform pixel-level classification of arbitrary objects in an image, even if these objects have not been present in the training data.

Let X represent an input image and Y represent the corresponding pixel-level category labels. Traditional semantic segmentation models learn a mapping function $f : X \to Y$, where Y belongs to a fixed category set C_{train}.

In OVSS, the goal is to learn a mapping function $f' : X \to Y'$, where Y' includes categories beyond those seen during training, represented as C_{open}. This means that during testing, the model should be able to segment new categories C_{test} such that $C_{\text{test}} \cap C_{\text{train}} = \emptyset$, yet f' should still accurately segment C_{test}.

2.2. History and Roadmap

The earliest work utilizing neural networks for semantic segmentation was proposed by Jonathan Long et al. in 2015 [18]. Their work was the first to introduce the conversion of traditional CNNs into Fully convolutional networks(FCN) for end-to-end pixel-level semantic segmentation, regarding as a cornerstone of modern semantic segmentation research, establishing the foundation for subsequent models such as U-Net [21] and Deeplab [5]. However, these traditional semantic segmentation models depend on a predefined set of fixed categories during the training phase and are incapable of classifying categories beyond the training set.

With the rise of BERT [7] in natural language processing (NLP), multimodal pretraining has garnered significant attention. Inspired by visual-language pretraining, OVR-CNN [27] introduced the concept of open-vocabulary object detection, utilizing caption data to connect novel category semantics with visual regions. Subsequently, CLIP [20] enabled models to perform zero-shot classification (ZSC) and recognition on categories that were not explicitly included during training. LSeg [16] was the first to investigate the application of CLIP in language-driven segmentation tasks.OpenSeg [11] proposed shifting from pixel-level feature matching to region-level feature matching, adapting the model to a two-stage structure. Building on these foundational works, OVSeg [17] significantly improved the performance of two-stage region-level image-text matching. SAM [15] introduced the concept of a segmentation foundation model, training on billions of masks. When combined with CLIP, SAM can achieve robust zero-shot segmentation(ZSS) results without requiring fine-tuning.

Today, with the rapid development of large language models (LLM), OVSS has become an increasingly promising research direction in the field of computer vision.

2.3. Preliminaries

2.3.1 Traditional Close-Set Semantic Segmentation

FCN [18] approaches the semantic segmentation problem as a dense pixel classification task. Following FCN, numerous studies have sought to improve and expand upon its framework. For instance, DeepLab [5] enhanced FCN's performance using dilated convolutions, conditional random fields and atrous spatial pyramid pooling. U-Net [21], on the other hand, improved multi-scale feature extraction through multi-scale feature fusion.

With the advent of Transformers [22] in 2017, some research [10] has proposed variants of self-attention mechanisms to better model global context, replacing the traditional CNN prediction heads.

2.3.2 Vision-Language Pre-Trained Models(VLPMS)

VLPMS represent a significant advancement in both computer vision and natural language processing in recent years. These models are trained on joint image and text data, learning to map images and texts into a common embedding space.

- **CLIP**. Proposed by OpenAI, CLIP [20] uses contrastive learning to align natural language with images, enabling pretraining on large-scale image-text datasets. CLIP employs a text encoder and an image encoder to embed both text and images into a shared vector space, ensuring that the embedding vectors of semantically similar text and images are close to each other. This model can understand and classify previously unseen image categories using textual descriptions, without requiring task-specific datasets.
- **ALIGN**. The ALIGN [13] model is trained on a massive dataset of images and their corresponding text descriptions using a contrastive learning approach similar to CLIP. ALIGN's training involves billions of image-text pairs, endowing the model with strong generalization and open vocabulary recognition capabilities.

The core advantage of VLPMS lies in their ability to infer and classify new categories without explicit labeling. This makes them particularly useful for OVSS tasks, as they can leverage natural language descriptions to identify and segment new image categories without retraining the model.

2.3.3 Transfer Learning and Self-Supervised Learning

Transfer learning and self-supervised learning are essential techniques for enhancing a model's generalization ability, enabling it to adapt to new categories and data distributions.

- **Transfer Learning**. Transfer learning involves applying a model pre-trained on a large-scale dataset to a new task, typically requiring only fine-tuning. In semantic segmentation tasks, the weights of pre-trained vision-language models (such as CLIP and ALIGN) can be used as initial weights, followed by fine-tuning on the target dataset. This approach not only accelerates the training process but also significantly improves performance on new tasks.
- **Self-Supervised Learning**. Self-supervised learning involves designing pretext tasks that allow a model to train without manual annotations. Examples include DINO [3] and MAE [12]. DINO employs a self-distillation method, where the model learns consistent features from different views of the same image. MAE involves masking parts of an image and training the model to reconstruct these parts, thereby learning both global and local image features. These self-supervised learning methods can train on large-scale unlabeled data, enhancing the model's generalization ability.

2.3.4 Parameter-Efficient Fine-Tuning(PEFT)

PEFT techniques aim to reduce the number of parameters that need to be updated during transfer learning, thereby increasing the efficiency and effectiveness of model fine-tuning. When fine-tuning large pre-trained models, adjusting only a small number of parameters can achieve good performance, avoiding the high computational costs and memory requirements associated with full model fine-tuning.

2.4. Datasets and Metrics

Datasets. For semantic segmentation, the most commonly used datasets are Pascal VOC [9], COCO Stuff [1], ADE20K-150, ADE20K-847 [28], Pascal Context-59 and Pascal Context-459 [19].

Metrics. The commonly used metric is mean intersection over union (mIoU), which measures the overlap between the predicted results and the ground truth. mIoU is calculated as follows:

$$\text{mIoU} = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FP_i + FN_i} \tag{1}$$

where TP_i (True Positives) is the number of pixels predicted to be of class i that are actually of class i, FP_i (False Positives) is the number of pixels predicted to be of class i but are not actually of class i, and FN_i (False Negatives) is the number of pixels that are actually of class i but are predicted to be of another class.

3. Methods

In this section, we will categorize the models into four types based on their architecture: Pixel-Level Feature Matching, Two-Stage Region-Level Image-Text Matching, Improved Single-Stage, and Open Segmentation. We will progressively introduce representative models from these four categories, following their technical routes and logical hierarchies, and highlight the research motivations and key technical points.

3.1. Pixel-Level Feature Matching

LSeg

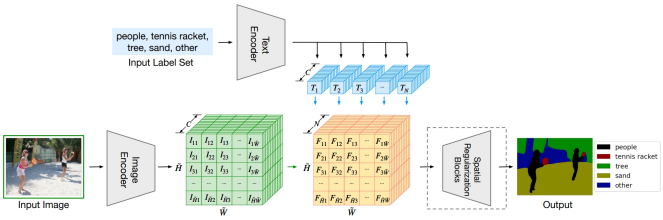

Figure 1. Overview of the LSeg framework. A text encoder converts labels into vectors. An image encoder creates per-pixel embeddings from the image and matches each pixel's feature to all label embeddings. The image encoder is trained to enhance the match between the text embedding and the correct image pixel embedding. A final spatial regularization block refines and cleans up the predictions.

Language-driven Semantic Segmentation (LSeg) [16] was the first to explore the application of CLIP in language-driven segmentation tasks and to introduce the concept of open vocabulary into semantic segmentation. LSeg replaces the traditional closed vocabulary classifier with an open classifier based on image-text feature matching, such as CLIP[20], and proposes pixel-level feature matching to achieve pixel-text feature alignment. Specifically, LSeg uses a text encoder to compute embeddings of descriptive labels and a Transformer-based image encoder to compute dense per-pixel embeddings of input images. The image encoder is trained with contrastive learning objectives to align pixel embeddings with the corresponding semantic category text embeddings. Text embeddings provide a flexible label representation, where semantically similar labels are mapped to similar regions in the embedding space. This allows LSeg to generalize to previously unseen categories during testing without requiring retraining or additional training samples.

The algorithm flow of LSeg is shown in Figure 1. Specifically, the process begins with using a frozen-parameter CLIP text encoder to extract category text features. The input category label set generates N text embedding vectors $T \in \mathbb{R}^{N\times C}$ through the CLIP text encoder.

Next, the input image is processed by the Dense Prediction Transformers (DPT) image encoder to produce per-pixel image embeddings $I \in \mathbb{R}^{\tilde{H} \times \tilde{W} \times C}$.The CLIP text features and the image feature map are then used to compute the dot product similarity per pixel, resulting in a per-pixel correlation tensor $F = I \cdot T \in \mathbb{R}^{\tilde{H} \times \tilde{W} \times N}$.Finally, the prediction results undergo spatial regularization and upsampling through a spatial regularization module, producing the segmentation result map. Each pixel is assigned to the most similar label category, and cross-entropy loss is calculated per pixel for training.

The experimental results demonstrate that, under zero-shot settings, LSeg achieves significant improvements compared to existing methods. However, its segmentation performance still lags behind traditional few-shot methods. Overall, as a pioneering work in the field of OVSS, LSeg has inspired a new paradigm of pixel-level image-text matching to achieve open-vocabulary segmentation.

3.2. Two-Stage Region-Level Image-Text Matching

OpenSeg

Although the LSeg introduced the concept of open vocabulary into semantic segmentation, the main issue with models that perform direct pixel-level feature matching is the scalability of training data. Pixel-level matching requires pixel-level annotations, which are typically very difficult and expensive to obtain for open vocabulary labels. Moreover, direct pixel matching can result in confusion and unclear boundaries. To address these challenges, Golnaz Ghiasi et al. proposed OpenSeg [11]. This model first extracts mask regions and then performs region-level matching within these regions, shifting feature matching from the pixel level to the region level and thus transforming the model into a two-stage structure. The difference between region-level feature matching and pixel-level feature matching is illustrated in Figure 2.

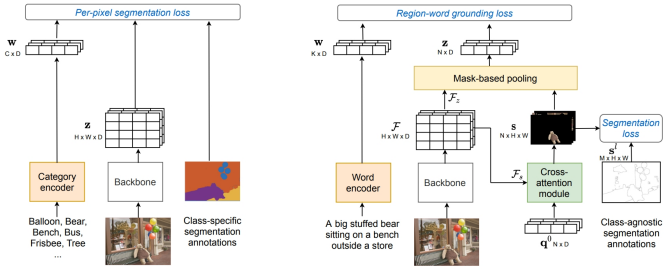

Figure 2. The difference between region-level feature matching and pixel-level feature matching.

Specifically, OpenSeg uses proposal masks and their features $Z \in \mathbb{R}^{N \times D}$ to represent images, enabling accurate image segmentation from image captions through weakly supervised learning. First, a feature pyramid network (FPN) and a cross-attention module extract multi-scale features F from the image, and enhanced image features F_{PEs} are obtained through convolution and fully connected layers. Masks are predicted by calculating the dot product of mask queries q and position-enhanced image features, resulting in $s = \text{Sigmoid}(\text{dot}(q, F_{PEs}))$. During optimization, mask matching is optimized by calculating the Dice coefficient between the predicted mask s and the unlabelled mask s_l, maximizing their similarity:

$$L_S = \frac{1}{M} \sum_{j=1}^{M} \left(1 - \max_i \text{Dice}(s_i, s_{lj})\right) \tag{2}$$

To achieve vision-semantic alignment, OpenSeg aligns image regions with words in the caption by calculating the cosine similarity between region features z and word features w and maximizing the normalized scores of annotated image-caption pairs. The similarity score between region i and word j is defined by the cosine similarity $\langle z_i, w_j \rangle = \frac{z_i \cdot w_j}{\|z_i\| \|w_j\|}$. The similarity calculation for image I_b and caption C_b is:

$$G(I_b, C_b) = \frac{1}{K} \sum_{j=1}^{K} \sum_{i=1}^{N} \sigma(g(z, w_j))_i \cdot \langle z_i, w_j \rangle \tag{3}$$

where σ represents the softmax function. The alignment loss aims to maximize the normalized scores of annotated image-caption pairs among all images and captions:

$$L_G = -\frac{1}{|B|} \sum_{b=1}^{|B|} (\log \sigma(G(I, C_b))_b + \log \sigma(G(I_b, C))_b) \tag{4}$$

To expand the training data, OpenSeg employs a self-training method. First, a teacher model is trained on a segmentation dataset using only the segmentation loss L_S. The teacher model then generates pseudo-segmentation labels for a large-scale image-text dataset, and the final model is trained on a mix of real and pseudo labels. The final total loss is:

$$L = L_G + \alpha L_S \tag{5}$$

This approach enables the model to handle any number of categories in open vocabulary segmentation tasks and provides high-quality segmentation predictions.OpenSeg has demonstrated superior performance across multiple datasets, particularly in zero-shot and few-shot settings. Furthermore, it exhibits strong generalization capabilities for previously unseen categories.

ZegFormer

Similar to OpenSeg, Jian et al. also improved region-level feature matching[8]. They proposed ZegFormer for the

zero-shot semantic segmentation (ZS3) task, decoupling the segmentation problem into two stages: class-agnostic mask generation and mask classification. Additionally, they introduced a vision-language model to improve mask classification accuracy by classifying the mask map obtained from the original image.

ZegFormer first generates a set of segment-level embeddings and performs class-agnostic grouping and segment-level zero-shot classification through two parallel layers. MaskFormer is chosen as the base semantic segmentation model, generating segment embeddings $G_q \in \mathbb{R}^d$ and mask embeddings $B_q \in \mathbb{R}^d$ by inputting N segment queries and a feature map into the Transformer decoder.

In the class-agnostic grouping stage, ZegFormer uses binary mask prediction to group the feature map $F(I) \in \mathbb{R}^{d\times H\times W}$ output by the pixel decoder, predicting masks $m_q = \sigma(B_q \cdot F(I)) \in [0,1]^{H\times W}$. During segment classification using SSE, class names are placed into a prompt template and input into the text encoder to obtain text embeddings $T = \{T_c \in \mathbb{R}^d | c = 1, \ldots, |C|\}$, and the segment prediction probability distribution is calculated through cosine similarity:

$$p_q(c) = \frac{\exp(\frac{1}{\tau}\text{sc}(T_i, G_q))}{\sum_{i=0}^{|C|}\exp(\frac{1}{\tau}\text{sc}(T_i, G_q))} \tag{6}$$

where $\text{sc}(e, e') = \frac{e\cdot e'}{|e||e'|}$, and τ is the temperature parameter.

During training, the mask loss $L_{\text{mask}}(m_q, R_{gt_q})$ is calculated using a combination of Dice loss and Focal loss. During inference, the predicted binary masks and segment classification scores are integrated to obtain the final results, and three variants of ZegFormer are proposed:

- ZegFormer-seg: Uses the segment classification scores of the segment queries, calculating the class probability for each pixel:

$$\sum_{q=1}^{N} p_q(c) \cdot m_q[h, w] \tag{7}$$

  Calibrates the prediction by reducing the scores of seen classes, with the final class prediction for each pixel being:

$$\arg\max_{c\in S+U}\left(\sum_{q=1}^{N} p_q(c) \cdot m_q[h, w] - \gamma \cdot I[c \in S]\right) \tag{8}$$

  where $\gamma \in [0, 1]$ is a calibration factor and the indicator function I is 1 if c belongs to seen classes.
- ZegFormer-img: The inference process is similar to formula (2), with the only difference being that $p_q(c)$ is replaced by $p'_q(c)$.
- ZegFormer: Integrates $p_q(c)$ and $p'_q(c)$:

$$p_{q,\text{fusion}}(c) = \begin{cases} p_q(c)^{1-\lambda} \cdot p_{q,\text{avg}}^{\lambda} & \text{if } c \in S \\ p_q(c)^{1-\lambda} \cdot p'_q(c)^{\lambda} & \text{if } c \in U \end{cases} \tag{9}$$

  where λ balances the contributions of the two classification scores. When c belongs to S, the geometric mean of $p_q(c)$ and $p_{q,\text{avg}} = \sum_{j\in S} p'_q(j)/|S|$ is calculated. The final semantic segmentation result is obtained through a process similar to formula 8.

By integrating different classification scores and adjusting the probability ranges of seen and unseen classes, ZegFormer achieves more accurate semantic segmentation.

OVSeg

After the introduction of models like ZegFormer, which are based on two-stage region-level image-text matching, many researchers have sought to further improve the performance of these models. Feng et al. [17] discovered that, due to the significant distribution differences between CLIP's training data and masked images, the combination of "Candidate Region Generation and Ground Truth Classification" significantly outperforms "Ground Truth Region and Predicted Classification." This indicates that the performance bottleneck of two-stage open-vocabulary semantic segmentation methods lies in region mask classification.

OVSeg enhances model performance by fine-tuning CLIP. First, suitable CLIP region fine-tuning data based on global image-text pairs is constructed: MaskFormer is trained using existing supervised data, and candidate masks are extracted to generate masked images. Then, category names are extracted from captions, and CLIP calculates the similarity between the masked images and the category names, assigning pseudo-labels to the best-matching categories.

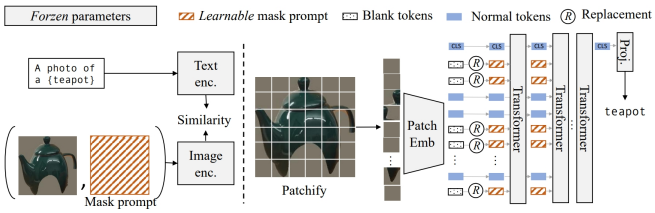

Figure 3. The Overview of Mask Prompt Tuning.

After constructing the data, Mask Prompt Tuning is used to fine-tune CLIP for masked image inputs. As shown in Figure 3, MaskFormer is first trained with existing annotated data, and candidate masks are extracted to generate masked images. Then, category names are extracted from the captions, and CLIP calculates the similarity between the masked images and the category names, assigning pseudo-labels to the most similar categories. To adapt CLIP to masked image input, a learnable mask prompt token is added, replacing patch embeddings that do not contain masked regions. During fine-tuning, the loss function remains consistent with CLIP, but only the mask prompt

token is updated, with CLIP's original parameters frozen. This process generates masked image data with pseudo-labels, which is used to fine-tune the CLIP model and improve its classification performance on masked images.

3.3. Improved Single-Stage

SAN

Because segmentation datasets are much smaller than vision-language pretraining datasets, fine-tuned models often have limited capabilities in open vocabulary recognition. Fine-tuning VLPMS like CLIP can affect their generalization ability, weakening their predictions for new classes. Additionally, the two-stage method of generating mask images and then using CLIP for classification requires multiple inferences with CLIP. Since the mask prediction model is independent of the VLPMS, it misses the opportunity to utilize the pretrained model's powerful features. This can result in unsuitable mask image crops, making the model cumbersome, slow, and underperforming. Based on this, Xu et al. [24] proposed an efficient open vocabulary semantic segmentation framework called Side Adapter Network (SAN) that requires no fine-tuning and only one inference.

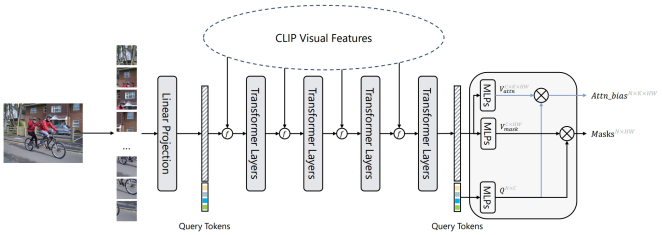

Figure 4. Overview of the SAN framework.

As illustrated in the Figure 4, SAN introduces a side adapter network based on CLIP to generate masks and mask attention, guiding CLIP to focus more on the mask regions and improving classification performance. SAN uses a lightweight vision transformer. The input image is divided into 16×16 patches, projected into visual tokens through a linear embedding layer, and then connected with learnable query tokens before being input into the transformer layers. SAN generates mask proposals and corresponding attention biases for mask recognition. The mask is generated by the inner product of query tokens and visual tokens:

$$M = V_{mask} Q_{mask}^T \tag{10}$$

Attention biases are generated similarly and applied to CLIP's self-attention layers:

$$B = V_{attn} Q_{attn}^T \tag{11}$$

This decoupled design allows the regions of interest for mask recognition to differ from the mask regions themselves, thus improving performance.

To further leverage CLIP's powerful features, SAN performs feature fusion by integrating CLIP's visual tokens into SAN. By fusing the features of CLIP and SAN layer by layer, model performance is significantly improved. Additionally, SAN introduces shadow [CLS] tokens ([SLS] tokens) to guide the attention map of the [CLS] token through attention biases without altering the CLIP model parameters, achieving more precise mask recognition.

During training, SAN supervises mask generation with dice loss L_{mask_dice} and binary cross-entropy loss L_{mask_bce}, and supervises mask recognition with cross-entropy loss L_{cls}. The total loss is the weighted sum of these losses:

$$L_{seg} = \lambda_1 L_{mask\_dice} + \lambda_2 L_{mask\_bce} + \lambda_3 L_{cls} \tag{12}$$

where λ_1, λ_2, and λ_3 are loss weights. Through end-to-end training, the side adapter network can maximally adapt to the frozen CLIP model, making the mask proposals and attention biases CLIP-aware.

Deep features are more semantic than shallow features, and multi-layer feature fusion improves performance more than single-layer fusion. Additionally, SAN adopts a single-pass design, with shallow CLIP layers used for feature fusion and deep CLIP layers used for mask recognition, reducing computational costs and significantly outperforming existing methods in multiple benchmarks.

FC-CLIP

In addition to SAN, Yu et al. have developed another single-stage OVSS called FC-CLIP [26]. FC-CLIP unifies mask generation and classification by sharing a frozen CLIP convolutional backbone. In traditional two-stage methods, a mask generator first creates candidate mask regions, which are then classified by the CLIP model, requiring separate feature extractions for masks and text, leading to inefficiency. FC-CLIP, however, uses a shared frozen CLIP convolutional backbone, allowing simultaneous use of image features for both mask generation and classification, avoiding redundant feature extraction. This backbone retains the pretrained image-text alignment properties while generating high-quality masks and accurate classifications, greatly improving efficiency and performance.

Using a shared, frozen convolutional CLIP backbone, FC-CLIP shows better generalization on high-resolution images needed for dense predictions compared to ViT-based CLIP models (like SAN). The convolutional model produces smoother features, making it more suitable for mask-pooling to extract mask region features. Additionally, FC-CLIP maintains very low training and testing costs while

significantly outperforming contemporary methods in accuracy.

3.4. Open Segmentation

SAM and its variants

The Segment Anything Model (SAM) [15] developed by Meta AI is a cutting-edge AI model designed for promptable semantic segmentation. It excels at segmenting objects within images with a single click, demonstrating zero-shot generalization to unfamiliar objects and images without requiring additional training. This capability is enabled by its architecture, which includes an image encoder, a prompt encoder, and a mask decoder working together to deliver precise segmentation outputs (Segment Anything).

SAM leverages advanced technologies such as CNNs and Generative Adversarial Networks (GANs) to perform detailed image analysis and generate accurate segmentation masks. CNNs help in recognizing and interpreting patterns in images, while GANs enhance the model's ability to create lifelike and precise segmentations. This combination allows SAM to handle a wide array of visual inputs with high precision, making it a significant advancement in the field of image segmentation.

A key factor in SAM's effectiveness is its extensive training on the SA-1B dataset, which includes over 1 billion segmentation masks from 11 million images. This vast and diverse dataset ensures that SAM can generalize well across various tasks and environments, enhancing its applicability in numerous domains such as AI-assisted labeling, medical imaging, and land cover mapping.

Due to the "Segment Everything" capability of SAM, it is particularly well-suited for generalization and transfer to various downstream tasks. To date, numerous models based on SAM have been developed and improved. Representative advancements include SEEM [25], a fast, interactive model for segmenting everything in an image simultaneously; SA3D [4], designed for 3D scene segmentation; SegGPT [23], a universal model for segmenting everything in context; and HQ-SAM, which focuses on improving mask quality [14].

4. Challenges And Outlook

4.1. Challenges

Base class overfitting problem. Most methods rely heavily on base class annotation data to detect and segment objects. However, there is a natural gap in distribution and semantic information between new classes and base classes, affecting the accuracy of segmenting new class objects. Although vision-language models (VLMs) can help bridge this gap through pretraining that captures knowledge between vision and text, most detectors still tend to overfit to base classes when new and base class objects are very similar in shape and semantics. To address this overfitting problem, finer-grained feature discrimination modeling is needed.

Better benchmarks and metrics. The current datasets are still relatively small and insufficient for OVSS-related tasks. Meanwhile, the quality of many existing datasets still has issues, and more standardized and reasonable metrics need to be proposed to better evaluate open vocabulary methods.

Training cost. Most state-of-the-art methods require a large amount of data for pretraining to achieve good performance. These costs can be expensive or even unaffordable for many researchers. Although using a frozen backbone network can help mitigate these costs, it may impact the model's performance.

4.2. Future Directions

Real-time open vocabulary detection and segmentation. Current models are large and slow, making real-time applications challenging. To fully realize the potential of open vocabulary segmentation, exploring real-time semantic segmenters with open vocabulary recognition capabilities is a promising research direction.

Combining large language models. Compared to VLMs, most LLMs contain a broader range of text concepts, naturally covering more than various dataset taxonomies. Thus, better aligning LLM knowledge with visual segmenters to achieve stronger zero-shot results remains an area for exploration.

Unifying open vocabulary tasks. Unification is an inevitable trend in computer vision. Currently, no model can be applied to all open vocabulary tasks and datasets. Developing a universal foundational model for the open vocabulary domain is a research-worthy direction.

5. Conclusion

In this survey, we have thoroughly examined the field of open vocabulary visual segmentation(OVSS). In the background section, we briefly introduced the problem definition, history, preliminaries, datasets, and metrics of OVSS. In the methods section, we categorized the models into four types based on their architecture: Pixel-Level Feature Matching, Two-Stage Region-Level Image-Text Matching, Improved Single-Stage, and Open Segmentation. We then progressively introduced representative models from these four categories according to their technical routes and logical hierarchies, highlighting the research motivations and key technical points. Finally, we summarized the existing challenges and suggested future research directions for open vocabulary learning.

References

[1] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *CVPR*, 2018. 3

[2] Senay Cakir, Marcel Gauß, Kai Häppeler, Yassine Ounajjar, Fabian Heinle, and Reiner Marchthaler. Semantic segmentation for autonomous driving: Model evaluation, dataset generation, perspective comparison, and real-time capability. *arXiv preprint arXiv:2207.12939*, 2022. 1

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 3

[4] Jiazhong Cen, Jiemin Fang, Zanwei Zhou, Chen Yang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Segment anything in 3d with radiance fields. *arXiv preprint arXiv:2304.12308*, 2023. 7

[5] 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. *TPAMI*, 2017. 2

[6] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 1

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv*, 2018. 2

[8] Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. In *CVPR*, 2022. 4

[9] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 2015. 3

[10] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019. 2

[11] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *ECCV*, 2022. 2, 4

[12] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 3

[13] 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 *ICML*, 2021. 2

[14] Lei Ke, Mingqiao Ye, Martin Danelljan, Yu-Wing Tai, Chi-Keung Tang, Fisher Yu, et al. Segment anything in high quality. *Advances in Neural Information Processing Systems*, 36, 2024. 7

[15] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv*, 2023. 2, 7

[16] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *arXiv*, 2022. 1, 2, 3

[17] 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 *CVPR*, 2023. 2, 5

[18] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1, 2

[19] 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 *CVPR*, 2014. 3

[20] 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 *ICML*, 2021. 1, 2, 3

[21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 1, 2

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017. 2

[23] Xinlong Wang, Xiaosong Zhang, Yue Cao, Wen Wang, Chunhua Shen, and Tiejun Huang. Seggpt: Segmenting everything in context. *arXiv preprint arXiv:2304.03284*, 2023. 7

[24] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. Side adapter network for open-vocabulary semantic segmentation. In *CVPR*, 2023. 6

[25] Hao Zhang Feng Li Linjie Li Jianfeng Wang Lijuan Wang Jianfeng Gao Yong Jae Lee Xueyan Zou, Jianwei Yang. Segment everything everywhere all at once. *arXiv*, 2023. 7

[26] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. *NeurIPS*, 2024. 6

[27] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *CVPR*, 2021. 2

[28] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 3