Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.060295

ARTICLE

CPEWS: Contextual Prototype-Based End-to-End Weakly Supervised Semantic Segmentation

Xiaoyan Shao[1], Jiaqi Han[1,*], Lingling Li[1,*], Xuezhuan Zhao[1,2,3,4] and Jingjing Yan[1]

[1]School of Computer Science, Zhengzhou University of Aeronautics, Zhengzhou, 450046, China
[2]National Key Laboratory of Air-Based Information Perception and Fusion, Luoyang, 471000, China
[3]Chongqing Research Institute of Harbin Institute of Technology, Chongqing, 401151, China
[4]Aerospace Electronic Information Technology Henan Collaborative Innovation Center, Zhengzhou, 401151, China

*Corresponding Authors: Jiaqi Han. Email: hanjiaqi457@zua.edu.cn; Lingling Li. Email: lilingling@zua.edu.cn

Received: 29 October 2024; Accepted: 02 January 2025; Published: 26 March 2025

ABSTRACT: The primary challenge in weakly supervised semantic segmentation is effectively leveraging weak annotations while minimizing the performance gap compared to fully supervised methods. End-to-end model designs have gained significant attention for improving training efficiency. Most current algorithms rely on Convolutional Neural Networks (CNNs) for feature extraction. Although CNNs are proficient at capturing local features, they often struggle with global context, leading to incomplete and false Class Activation Mapping (CAM). To address these limitations, this work proposes a Contextual Prototype-Based End-to-End Weakly Supervised Semantic Segmentation (CPEWS) model, which improves feature extraction by utilizing the Vision Transformer (ViT). By incorporating its intermediate feature layers to preserve semantic information, this work introduces the Intermediate Supervised Module (ISM) to supervise the final layer's output, reducing boundary ambiguity and mitigating issues related to **incomplete activation**. Additionally, the Contextual Prototype Module (CPM) generates class-specific prototypes, while the proposed Prototype Discrimination Loss ($\mathscr{L}_{PDL}$) and Superclass Suppression Loss ($\mathscr{L}_{SSL}$) guide the network's training, effectively addressing **false activation** without the need for extra supervision. The CPEWS model proposed in this paper achieves state-of-the-art performance in end-to-end weakly supervised semantic segmentation without additional supervision. The validation set and test set Mean Intersection over Union (MIoU) of PASCAL VOC 2012 dataset achieved 69.8% and 72.6%, respectively. Compared with ToCo (pre trained weight ImageNet-1k), MIoU on the test set is 2.1% higher. In addition, MIoU reached 41.4% on the validation set of the MS COCO 2014 dataset.

KEYWORDS: End-to-end weakly supervised semantic segmentation; vision transformer; contextual prototype; class activation map

1 Introduction

Weakly supervised semantic segmentation aims to train neural networks using weak labels to generate reliable pixel-level pseudo labels. Researcher only need to perform simple annotation on the dataset samples to obtain weak labels, which can greatly reduce annotation costs. The common types of weak labels include image-level labels [1], points [2], bounding boxes [3], and scribbles [4]. Image-level labels are particularly easy to obtain. Most studies use image-level labels for weakly supervised semantic segmentation, and the paper also uses image level labels to train the network. Most existing weakly supervised semantic segmentation methods follow two stages [5]. The first stage generates relatively accurate pixel-level pseudo labels, while the second stage uses these labels for model training. However, this two-stage process requires separate training

for each stage, making the overall training complex and inefficient. Therefore, the paper employs an end-to-end [6] training method, where a single model is established and trained throughout the entire process. This approach allows for simultaneous model optimization and refinement of pixel-level pseudo labels, leading to improved segmentation results.

CAM [7] is a heat map generated by neural networks, which can convert image-level labels into pixel-level activation maps, assigning an activation value to each pixel to represent the activation intensity of the class in the image. Most weakly supervised semantic segmentation methods based on image-level labels first generate the CAM as the initial pseudo-label and subsequently refined [8] to produce a more reliable pseudo-label for model training. However, CAM has inherent limitations in semantic localization. It typically only activates the most recognizable semantic regions, leading to problems with incomplete activation and false activation of unknown classes. As illustrated in Fig. 1a, CAM activates only the most recognizable regions, such as the cats', sheep's, and dogs' main body areas, while failing to fully activate the legs, resulting in **incomplete activation**. In Fig. 1b, the first row shows that CAM false associates branches with the bird's tail due to their similarity, highlighting a problem of **false activation**. Similar issues are observed with the aircraft and runway in the second row, and with seawater and ships in the third row. These problems significantly impair the performance of weakly supervised semantic segmentation models.

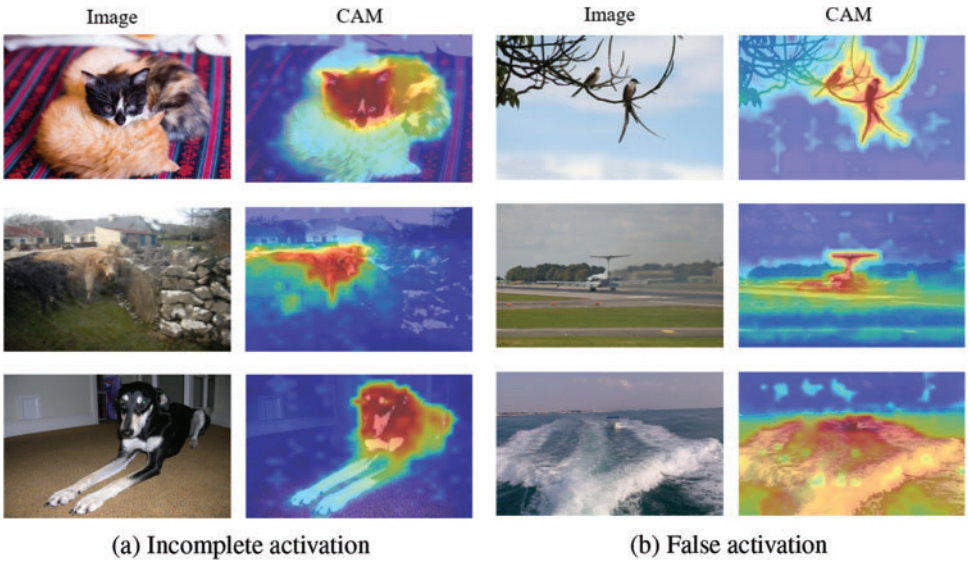

Figure 1: (a) Incomplete activation. CAM fails to fully activate all relevant regions, leading to incomplete coverage of some target regions. **(b) False activation**. CAM mistakenly activates regions unrelated to the target

Recent studies have shown that incomplete activation often arises because most methods use CNNs to generate CAM. CNNs rely on smaller convolution kernels for local feature extraction, which makes it challenging to capture global information. For instance, Fatima et al. [9] present a solution to enhance CNN performance by incorporating U-Net for saliency estimation and adding residual connections, which help capture richer feature information and improves segmentation accuracy. Unlike traditional methods, the ViT employs a self-attention mechanism, enabling global feature extraction across the entire sequence. Additionally, position encoding in ViT preserves the sequence's order and structural information, which enhances its ability to handle global information. For example, Ullah et al. [10] show that the self-attention mechanism in a lightweight ViT effectively captures global contextual information, enhancing the accuracy of recognition and classification for apple leaf diseases, while also demonstrating ViT's advantage in processing global

features. In contrast, traditional CNN-based methods are limited to local feature extraction. However, during feature extraction, object boundaries often become ambiguous, and adjacent patch tokens tend to exhibit increasing similarity. To address this, the paper improves feature extraction with ViT by utilizing an intermediate layer that preserves more semantic information and introducing an ISM. Specifically, ISM module inputs the intermediate layer features into a classifier to generate CAM, which then obtains cosine similarity relationships to supervise the high-level features. This helps generate more accurate pseudo labels, reducing boundary ambiguity and mitigating issues related to **incomplete activation**.

The problem of false activation has been investigated by some scholars [11] (detailed in relevant work). However, these methods heavily rely on additional supervision or human prior knowledge to identify categories with co-occurrence problems. This paper introduces a method based on class prototypes, adapted from the field of few-shot semantic segmentation, to achieve accurate segmentation without additional supervision (such as saliency maps, text information, additional supplementary data). In few-shot semantic segmentation, a single class representation prototype is typically used, but this often results in the activation of only a small number of pixels due to incomplete feature information extraction. In contrast, our integrated CPM method captures more accurate category features. By generating category-specific positive prototypes to represent the foreground and negative prototypes to represent the background using image-level labels, the paper address the issue of false activation. Furthermore, the paper propose $\mathscr{L}_{PDL}$ and $\mathscr{L}_{SSL}$ to better separate foreground and background, effectively suppressing **false activations** from background objects and generating more complete CAM.

This method achieved state-of-the-art results on the PASCAL VOC 2012 [12] and MS COCO 2014 [13] datasets. The key contributions of this work are as follows:

- This work proposes a novel end-to-end network model for weakly supervised semantic segmentation, named Contextual Prototype-Based End-to-End Weakly Supervised Semantic Segmentation (CPEWS), which stores contextual prototypes in a fixed pool of positive and negative prototypes to preserve their stability throughout the training process.
- By employing ViT as the backbone, this approach effectively captures global contextual information and models long-range dependencies across different regions of the image. The introduction of the ISM leverages the cosine similarity of intermediate layer features to supervise high-level features, reducing boundary ambiguity and mitigating incomplete activation.
- By employing the CPM, class-specific prototypes are generated using k-means clustering and class probability predictions. Based on these prototypes, the paper introduces two novel loss functions $\mathscr{L}_{PDL}$ and $\mathscr{L}_{SSL}$. These losses are designed to improve the separation between foreground and background, effectively reducing false activations from background objects.
- Extensive experiments on the PASCAL VOC 2012 and MS COCO 2014 datasets show that CPEWS achieves state-of-the-art performance in end-to-end weakly supervised semantic segmentation without additional supervision.

The rest of the paper is organized as follows. Section 2 primarily introduces the key techniques and advancements in the field of weakly supervised semantic segmentation, including end-to-end methods, incomplete activation, prototype learning, and false activations. Section 3 presents an overview of the proposed network framework, including the ISM and CPM modules. Section 4 primarily analyzes the experimental details. Section 5 summarizes the methods proposed in the paper, discusses potential future developments, and highlights the limitations of the proposed model.

2 Related Work

End-to-end weakly supervised semantic segmentation. Training a well-performing end-to-end weakly supervised semantic segmentation network is challenging due to the reliance on weak labels for supervision. Chen et al. [14] introduce a multi-granularity denoising module that addresses saliency maps noise and reduces the disparity between simple and complex data through a bidirectional alignment mechanism. Yang et al. [15] spatially separate co-occurring objects by subdividing the image into smaller blocks. In the feature space, semantic representation is enhanced through multi-granularity knowledge comparison, which effectively addresses the issue of false activations. Yang et al. [16] introduce uncertainty estimation to reduce bias and propose an affinity diversification module to foster greater semantic diversity. These methods excel in local feature extraction by leveraging CNN architectures, enabling them to efficiently capture fine-grained local information. However, the inherent limitation of CNNs lies in their constrained receptive fields, especially for tasks that require modeling long-range dependencies. To address this challenge in weakly supervised semantic segmentation, this paper introduces the ViT, which offers superior global feature modeling and effectively captures the relationships between different regions in the image.

Incomplete activation. ViT has shown strong performance in image processing tasks and has achieved significant breakthroughs in recent years. Recent research has begun integrating ViT into weakly supervised semantic segmentation tasks. Ru et al. [17] learn semantic affinity from Transformer's multi-head self-attention and design a Pixel-Adaptive Refinement module (PAR) combined with low-level visual information to further ensure local consistency of pseudo-labels. Sun et al. [18] capture class label attention by extracting gradients from attention maps and mapping other labels to their corresponding classes. Xu et al. [19] embed multiple class labels to enable the model to learn activation maps for different classes individually. Wu et al. [20] employ a dual-student framework with reliable progressive learning, leveraging discrepancy loss to generate multiple CAMs. These CAMs provide mutual supervision, helping to reduce incomplete activation caused by the learning of incorrect pseudo labels. These methods demonstrate the potential of ViT in weakly supervised semantic segmentation, where the attention mechanism and class labeling information can be effectively utilized to improve segmentation performance. However, these methods do not address the issue of boundary ambiguity that arises with ViT. In this paper, intermediate layer knowledge is used to supervise the features output by the last layer to improve boundary ambiguity and solve the problem of incomplete activation.

Prototype learning. Prototype learning has been extensively studied in few-shot semantic segmentation. The theory of prototype learning [21] demonstrates that prototypes can represent local features, global features, or specific properties of objects. Example prototypes [22] can dynamically represent the discriminant features of specific images, effectively handling intra-class variations in object features. Contextually integrated prototypes [23] can capture more specific and accurate categorical semantic patterns. Lang et al. [24] employ a two-stage training strategy. In the first stage, semantic segmentation is used to train extractors, and prototypes are generated through Mask Average Pooling (MAP). In the second stage, meta-learning is applied to integrate base classes with new classes. Kayaba et al. [25] build on this approach by adding multi-scale fusion, effectively addressing the issue of spatial inconsistency. Tang et al. [26] capture feature differences through contextual awareness and enhance model representation by aligning feature distributions. While these methods all leverage prototype learning and context-aware strategies to strengthen the expressive power of semantic segmentation, they incorporate prototypes as dynamic components of the learning model. In contrast, this paper stores prototypes in a fixed pool prior to training to ensure their stability. Additionally, loss functions are designed to optimize network parameters and enhance the model's ability to extract instance information.

False activation, also known as overactivation, refers to the false activation of the CAM in non-target regions. This problem has garnered significant attention in recent years. Chen et al. [27] optimize CAM to generate high quality pixel-level pseudo-labels by removing the co-occurrence relationship between categories in the image. Lee et al. [28] use saliency maps as pseudo pixel feedback to distinguish foreground objects from co-occurring backgrounds. Xie et al. [29] suppress background objects by inputting a series of preset background descriptions into the text encoder of the Contrastive Language-Image Pre-Training (CLIP) network and leveraging a pre-trained model. Lee et al. [30] enhance the network's ability to distinguish between targets and backgrounds by manually collecting additional training images containing co-occurring background objects. Zhang et al. [31] build on the CLIP model by using CLIP as a frozen backbone network for semantic feature extraction, while introducing a new decoder to interpret these features for prediction. These methods effectively address the challenge of separating foreground and background in weakly supervised semantic segmentation through innovative techniques. However, they heavily depend on additional supervision or human prior knowledge to identify categories with co-occurrence issues, which can result in false activations. In contrast, this paper's solution avoids the use of extra supervisory information or human knowledge. Instead, it addresses the foreground-background separation issue by incorporating contextual prototypes.

3 Methodology

The paper propose the CPEWS model for weakly supervised semantic segmentation, as illustrated in Fig. 2. This model employs ViT as the backbone to extract high-level features F_h and intermediate-level features F_m. ISM utilizes the intermediate layer features F_m to produce an intermediate activation map M^m, which supervises and constrains the features output of the final layer through a cosine similarity relationship. Additionally, CPM utilizes high-level features F_h to generate high-level activation maps M^h, which are then used to create positive and negative prototype representations for specific classes. Guided by these prototypes, a joint training optimization network is introduced, incorporating $\mathscr{L}_{PDL}$ and $\mathscr{L}_{SSL}$. This network effectively leverages the multi-layer features of the ViT model and achieves precise weakly supervised semantic segmentation through prototype learning and feature consistency.

3.1 Intermediate Supervised Module

The ViT is employed to effectively model global contextual information, capture long-range dependencies across different regions of the image, and leverage intermediate layer knowledge to reduce boundary ambiguity, thereby addressing the issue of **incomplete activation**. As shown in Fig. 2, the training data is fed into the ViT as an input sequence. The image X is flattened and linearly projected into tokens. In each Transformer block, multi-head self-attention is used to capture global feature dependencies. Specifically, for the ith head, the patch token is projected through an Multilayer Perceptron (MLP) layer to obtain the query $Q_i \in R^{hw \times d_k}$, key $K_i \in R^{hw \times d_k}$, and value $V_i \in R^{hw \times d_k}$. The $head_i$ output is generated based on Q_i, K_i, and V_i. This self-attention mechanism enables ViT to effectively capture global dependencies between image patches and extract more discriminative feature representations. All the formulas used in this paper are summarized in Appendix A.

$$Q_i = XW_i^Q,\ K_i = XW_i^K,\ V_i = XW_i^V \tag{1}$$

$$head_i = softmax\left(\frac{Q_i(K_i)^{\mathrm{T}}}{\sqrt{d_k}}\right)V \tag{2}$$

where, X represents the image sequence, W_i denotes the weight, Q_i, K_i, and V_i are the query, key and value matrixs obtained after the input vector passes through the linear projection layer, and d_k represents the

dimension of the key matrix. The final output of ViT's block is to concatenate the outputs of each attention head ($head_1, head_2, \ldots, head_h$), and then passing the concatenated result into the normalization layer and MLP layer for processing. Generate feature maps for subsequent modules by stacking multiple such blocks.

$$head = Concat\left(head_1, head_2, \ldots, head_h\right) \tag{3}$$

where, h represents the number of attention heads.

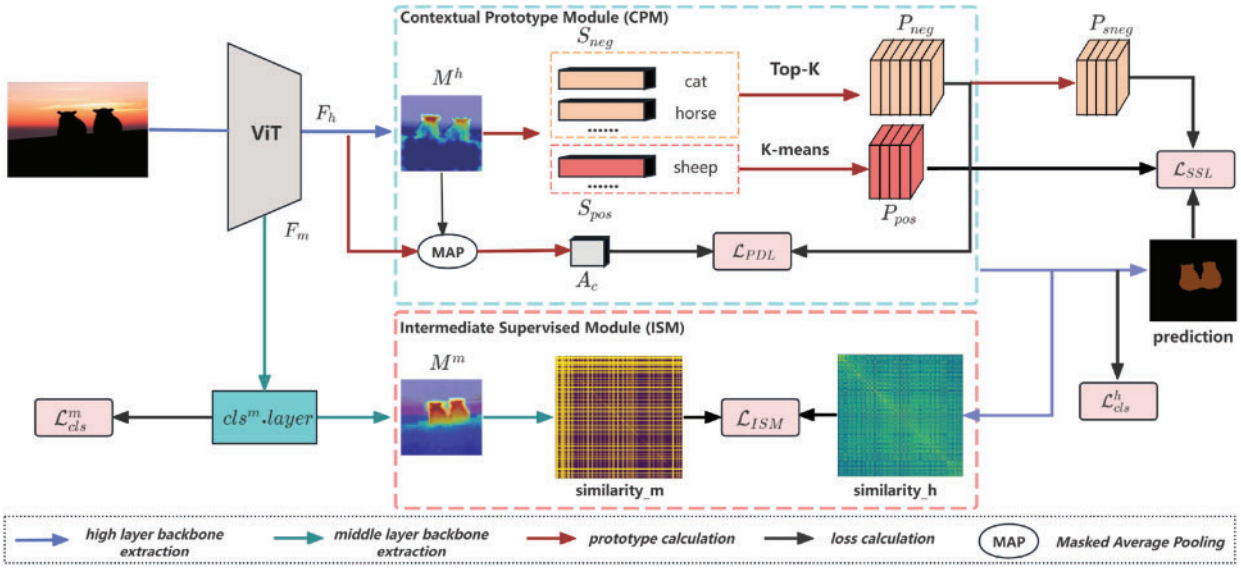

Figure 2: Overall framework of CPEWS. ISM introduces cosine similarity of intermediate layer features as an auxiliary supervision signal; CPM is used to generate positive and negative prototypes of a specific class, and to calculate the loss using the generated prototypes

Simultaneously, ISM is introduced. In the feature extraction process of the ViT backbone network, this method introduces a linear classification layer in the intermediate layer. Experiments indicate that, among the 12 blocks in ViT, the later layers retain more semantic information, while the earlier layers lack advanced feature capture. This linear classification layer generates an auxiliary class activation map M^m. Specifically, for each training image X and its image-level label $y = [y_1, y_2, \ldots, y_c] \in \{0,1\}^c$, the intermediate level features $F_m \in R^{D\times H\times W}$ are used to obtain the weight $W \in R^{c\times d}$ of the classification layer, where c represents the number of categories and d represents the number of channels. The corresponding weights are applied to the intermediate layer feature maps, and the weighted feature maps are summed to obtain the intermediate layer class activation map M^m. To ensure that negative values in CAM do not affect the final result, the ReLu activation function is applied. The CAM for class c is calculated as follows:

$$CAM_c = ReLu\left(\sum_{i=1}^{d} W^{i,c} F^i\right) \tag{4}$$

where, c denotes the category, d represents the dimensionality of the feature map, $W^{i,c}$ is the weight coefficient of the ith feature map corresponding to category c, and F^i represents the feature value of the ith feature map.

Subsequently, Using thresholds α and β (where $0 < \alpha < \beta < 1$), M^m is converted into pseudo-label Y. This pseudo-label Y is then used to establish precise cosine similarity relationships, which in turn supervise

the features of the final layer. Specifically, tokens with the same semantic label are classified as positive samples, while those with different labels are marked as negative samples. To address the boundary ambiguity problem in the final layer of ViT, this method aims to maximize the similarity between positive samples and the final layer features while minimizing the similarity between negative samples. The formula is as follows:

$$\mathscr{L}_{ISM} = \frac{1}{N^+} \sum_{Y_i = Y_j} (1 - CosSim(F_h^i, F_h^j)) + \frac{1}{N^-} \sum_{Y_i \neq Y_j} (1 - CosSim(F_h^i, F_h^j)) \tag{5}$$

where, $CosSim()$ function calculates the cosine similarity between tokens, N^+ and N^- denote the number of positive and negative samples, respectively. This loss function encourages greater consistency among features of positive sample pairs while enhancing the distinction between negative sample pairs. In summary, this method improves the diversity and discriminability of feature representations by minimizing the cosine similarity between positive and negative sample pairs. The algorithm's detailed framework is outlined in Algorithm 1.

Algorithm 1: ISM detailed structure

Input: Token sequence $X_{\text{input}} \in \mathbb{R}^{B \times N \times C}$
Initialize: ViT Backbone, Linear Layer for Intermediate Layer
Step 1: Generate Class Activation Maps (CAM)
for each image X and its label Y **do**
    Compute CAM for each class c:
$$\text{CAM}_c \leftarrow \text{ReLU}\left(\sum_{i=1}^{d} W^{i,c} F^i\right)$$
end for
Step 2: Pseudo-Label Generation
Use thresholds α, β to generate pseudo-labels Y
$Y \leftarrow \text{CAM}_c$
Step 3: Cosine Similarity Computation
for each token pair **do**
    $\text{CosSim}(F_h^i, F_h^j) \leftarrow$ Y
    positive pair ← same class
    negative pair ← different class
end for
Step 4: Loss Optimization
positive pair ← maximize similarity
negative pair ← minimize similarity
Output: Enhanced token sequence Y_{output}

3.2 Contextual Prototype Module

This module uses CPM to capture more accurate prototypes of specific classes. Specifically, it enhances the expressive power of category features by exploring the relationships between specific instances and other instances, constructing contextual prototypes. The high-level features F_h and CAM_c (selecting the activation map M^h obtained from the final layer) from Section 3.1 are used to calculate the predefined anchor frame

feature A_c. This feature embedding is obtained by applying MAP on M^h. The formula is shown as follows:

$$A_c = \frac{\sum_i CAM_c^i \cdot F^i}{\sum_i CAM_c^i} \tag{6}$$

where i is the pixel on the class localization map CAM_c and feature map F_h.

To store positive and negative samples, the paper establish the positive sample pool S_{pos} and the negative sample pool S_{neg} to initialize the network's feature extraction. Positive and negative samples are stored in $S_{pos} = [S_{pos}^1, S_{pos}^2, \ldots, S_{pos}^C]$ and $S_{neg} = [S_{neg}^1, S_{neg}^2, \ldots, S_{neg}^C]$, respectively. If class c appears in image X (i.e.,$y_c = 1$), the label y_c is added to S_{pos}; if class c does not appear in image X, then y_c is assigned to S_{neg}. By collecting all the images in the dataset, S_{pos} captures the intrinsic features of the foreground class, while S_{neg} gathers class-specific false features, including background information related to co-occurrence.

S_{pos} and S_{neg} typically contain a large amount of feature data. If these are simply used as training data, the network may struggle to effectively learn valuable features and require significant computational resources. To address this issue, a small number of representative feature representations are extracted from S_{pos} and S_{neg} for each class c. Class-specific positive prototypes $P_{pos}^c = [P_{pos}^{c,1}, P_{pos}^{c,2}, \ldots, P_{pos}^{c,K}]$ and negative prototypes $P_{neg}^c = [P_{neg}^{c,1}, P_{neg}^{c,2}, \ldots, P_{neg}^{c,K}]$ are constructed from S_{pos} and S_{neg}, respectively. The number of prototypes K is set to a small fixed value (e.g., 10) to reduce computational complexity.

The production processes for prototypes P_{pos}^c and P_{neg}^c are different. For P_{pos}^c, the k-means clustering algorithm is used to cluster the positive sample feature representations S_{pos}^c and obtain the cluster centroids. However, for P_{neg}^c, directly using k-means clustering may result in poor-quality negative prototypes due to the inclusion of some uncertain representations (e.g., $tracks, stations$). Therefore, instead of clustering, the representations in S_{neg}^c are sorted in descending order based on the predicted class probability $\hat{y_c}$, and the top K representations are selected as P_{neg}^c. Through these operations, more appropriate positive and negative prototypes for each feature type can be constructed. The algorithm's detailed framework is outlined in Algorithm 2.

Algorithm 2: CPM detailed structure

Input: Image X, Feature maps F, Class activation maps CAM
Initialize: Positive pool S_{pos}, Negative pool S_{neg}
Step 1: Compute context prototypes using CAM_c and F_h
feature embedding ← Masked Average Pooling (M^h)
Step 2: For each image X
for each class c **do**
    if class c is present in image X **then**
        $S_{\text{pos}}^{\text{c}} \leftarrow F_h$
    else
        $S_{\text{neg}}^{\text{c}} \leftarrow F_h$
    end if
end for
Step 3: For each class c
if positive class c **then**
    $P_{\text{pos}}^{\text{c}} \leftarrow \text{k-means}(S_{\text{pos}}^{\text{c}})$
else
    $P_{\text{neg}}^{\text{c}} \leftarrow \text{top-K}(S_{\text{neg}}^{\text{c}})$

(Continued)

Algorithm 2 (continued)

18: **end if**
19: **Step 4: Output Prototypes**
20: **Output:** positive and negative prototypes $(P_{\text{pos}}, P_{\text{neg}})$

3.3 Prototypical Discrimination Loss and Superclass Suppression Loss

To train the network with positive sample prototypes P^c_{pos}, negative sample prototypes P^c_{neg}, and predefined anchor box features A_c, the training process should focus only on the target regions where $y_c = 1$ in the training set. To guide the network in learning better feature representations, $\mathscr{L}_{PDL}$ is proposed. The goal of $\mathscr{L}_{PDL}$ is to bring the feature representation A_c closer to the center C^c_{pos} of the positive sample prototypes, reducing differences between similar targets, enabling the network to learn a more universal representation of target features, making the target features more compact. Simultaneously, the method pushes the feature representation A_c away from the negative sample prototype center C^c_{neg}, thereby enhancing the distinction between foreground targets and co-occurring background, allowing for better separation of foreground targets from the background. By minimizing $\mathscr{L}_{PDL}$, the network can learn more discriminative feature representations, thereby improving the final semantic segmentation performance. The formula for calculating $\mathscr{L}_{PDL}$ using cross entropy is as follows:

$$\mathscr{L}_{PDL} = -\frac{1}{\sum_{c=1}^{C} y_c} \sum_{c=1}^{C} y_c \cdot [log(R_{pos}(A_c, C^c_{pos})) + log(1 - R_{neg}(A_c, C^c_{neg}))] \tag{7}$$

where, the prototype P^c_{pos} and P^c_{neg} are averaged to obtain C^c_{pos} and C^c_{neg}, respectively. Based on the Euclidean theorem, the similarity R_{pos} and R_{neg} between P^c_{pos} and C^c_{pos}, as well as between P^c_{neg} and C^c_{neg}, are calculated. A similarity closer to 1 indicates greater similarity. The formula for calculating similarity is as follows:

$$R_{pos}(A_c, C^c_{pos}) = exp - \frac{\left\| A_c - C^c_{pos} \right\|_2}{\mathscr{T}} \tag{8}$$

$$R_{neg}(A_c, C^c_{neg}) = exp - \frac{min\left\| A_c - C^c_{neg} \right\|_2}{\mathscr{T}} \tag{9}$$

where, the parameter $\mathscr{T}$ is used to scale the Euclidean distance, allowing for adjustment of the distinction between foreground and background.

$\mathscr{L}_{PDL}$ increases the distance between foreground and co-occurring background, enhancing their separation. However, since the last layer of the network does not receive gradient information from $\mathscr{L}_{PDL}$, the linear classification layer lacks guidance on which feature representations to emphasize or filter. the model cannot adjust parameters based on the $\mathscr{L}_{PDL}$ information. To address this problem, an $\mathscr{L}_{SSL}$ is proposed. For each pixel activated in the CAM, $\mathscr{L}_{SSL}$ calculates whether the distance between the pixel and the nearest negative sample prototype is less than the distance to the nearest positive sample prototype. If the distance is smaller, the output of the pixel is suppressed. This approach helps the network better differentiate between foreground and background by directly suppressing outputs that resemble the background, thus improving overall classification performance. Unlike $\mathscr{L}_{PDL}$, $\mathscr{L}_{SSL}$ directly influences the final layer classifier, allowing it to better distinguish between foreground and background. Use Φ to represent the pixel-level collection that is suppressed in the cth CAM. The formula is as follows:

$$\Phi_c = \{ i \mid min(\| F^i - P^c_{neg} \|_2) < min(\| F^i - P^c_{pos}) \|_2 \} \tag{10}$$

The $\mathscr{L}_{SSL}$ formula is as follows:

$$\mathscr{L}_{SSL} = \frac{1}{\sum_{c=1}^{C} y_c \cdot \mid \Phi_c \mid} \sum_{c=1}^{C} y_c \cdot \sum_{i \in \Phi_c} CAM_c^i \tag{11}$$

where, CAM_c^i is the output value of pixel i in CAM_c.

Although this loss effectively suppresses co-occurrence background, some foreground information is also suppressed. For example, the front of a bus may be collected as false information from the front of a train, which can lead to the network suppressing the information from the front of the train. To solve this problem, a superclass information is proposed, which leverages the information provided by a known dataset to filter out shared negative prototypes and reconstruct class relationships. Specifically, when calculating $\mathscr{L}_{SSL}$ for a certain class, only the negative sample prototypes P_{sneg}^c that do not belong to the same superclass in the image are retained, and the Φ pixel set is updated accordingly. The formula is as follows:

$$\Phi_c = \{ i \mid min(\parallel F^i - P_{sneg}^c \parallel_2) < min(\parallel F^i - P_{pos}^c) \parallel_2 \} \tag{12}$$

By jointly optimizing $\mathscr{L}_{PDL}$ and $\mathscr{L}_{SSL}$, the model effectively addresses the problem of **false activations**. $\mathscr{L}_{PDL}$ assists in correcting the activation of irrelevant regions, while $\mathscr{L}_{SSL}$ refines the prototype features. This combination enables the model to more accurately capture the target object's region and produce more precise CAM.

3.4 Network Optimization

As illustrated in Fig. 2, the CPEWS model is comprised of five optimization loss functions: classification loss ($\mathscr{L}_{cls}^m$ and $\mathscr{L}_{cls}^h$), the Intermediate Supervised Module $\mathscr{L}_{ISM}$, Prototypical Discrimination Loss $\mathscr{L}_{PDL}$, and Superclass Suppression Loss $\mathscr{L}_{SSL}$. For classification loss, the features extracted from the middle and final layers of ViT are aggregated through Global Maximum Pooling to create a more compact feature representation. These aggregated features are then passed through a convolutional classifier to generate a class score vector $\hat{y}$. To optimize the classification results, a multi-label soft margin loss is employed to calculate the classification loss $\mathscr{L}_{cls}^m$ and $\mathscr{L}_{cls}^h$. The $\mathscr{L}_{ISM}$ uses the cosine similarity of ViT intermediate layer features to supervise high-level feature learning, helping to alleviate boundary ambiguity and resolve the issue of incomplete activation. The $\mathscr{L}_{PDL}$ leverages cross-entropy to improve semantic segmentation performance, while the $\mathscr{L}_{SSL}$ effectively suppresses incorrect background activations by comparing the distance between pixels and both positive and negative sample prototypes. In summary, the optimization of the CPEWS model is achieved through a linear combination of the losses mentioned above:

$$\mathscr{L} = \mathscr{L}_{cls}^m + \mathscr{L}_{cls}^h + \lambda_1 \mathscr{L}_{ISM} + \lambda_2 \mathscr{L}_{PDL} + \lambda_3 \mathscr{L}_{SSL} \tag{13}$$

where, λ_1, λ_2 and λ_3 are the weights of the loss function, respectively, 12×10^{-2}, 15×10^{-5} and 2×10^{-1}.

4 Experiments

4.1 Experimental Settings

Dataset. This paper utilizes widely used datasets, PASCAL VOC 2012 and MS COCO 2014, in the field of weakly supervised semantic segmentation. PASCAL VOC 2012 features 21 semantic categories (including background) and is often extended with the SBD dataset. The expanded dataset includes 10,582 training images, 1449 validation images, and 1464 test images, covering 20 common categories such as humans, animals, vehicles, and daily necessities. MS COCO 2014, a more challenging and large-scale dataset,

encompasses 81 categories (including background) across people, animals, transportation, and electronic products. This dataset consists of 83 K images for training, 40 K images for validation, and 41 K images for testing.

Evaluation indicators. Effective evaluation metrics are essential for assessing the performance of weakly supervised semantic segmentation models. Commonly used indicators include execution time and Mean Intersection over Union (MIoU). Execution time is an important metric, but hardware configuration can influence this factor. As a result, many weakly supervised semantic segmentation studies do not provide detailed results. MIoU is commonly used for the accuracy of weakly supervised semantic segmentation models, and the specific calculation formula is:

$$MIoU = \frac{1}{C+1}\sum_{i=0}^{c}\frac{G \cap P}{G \cup P} \tag{14}$$

where, G denotes the ground truth, P denotes the predicted result, C represents the total number of categories.

Experimental conditions. The experimental conditions in this study includes 48 GB of RAM, an NVIDIA GeForce RTX 4090 with 24 GB. The software environment consists of Python 3.9, PyTorch 1.12.1+cu116, torchaudio 0.12.1+cu116, and torchvision 0.13.1+cu116.

Parameter settings. The experiment employs the ViT-base (ViT-B) [32] as the backbone network, initialized with ImageNet pre-trained weights [33]. The PolyWarmupAdamW optimizer is used for training, with the following parameter settings: momentum set to 9×10^{-1}, weight decay factor set to 1×10^{-2}, network warmup for 2000 iterations, and a warmup learning rate of 1×10^{-6}. For the PASCAL VOC 2012 dataset, the training network is iterated 20,000 times with a batch size of 4. The thresholds α and β (as detailed in Section 3.1) are set to 25×10^{-2} and 7×10^{-1}, respectively. The number of prototypes K (as described in Section 3.2) is set to 10, the parameter $\mathscr{T}$ in Eqs. (8) and (9) is set to 13, and the weight parameters λ_1, λ_2 and λ_3 in Eq. (13) are set to 12×10^{-2}, 15×10^{-5} and 2×10^{-1}, respectively. During inference, multi-scale testing and intensive Conditional Random Field (CRF) post-processing are applied. For the MS COCO 2014 dataset, the network undergoes 80,000 iterations with a batch size of 4. The thresholds α and β are set to 25×10^{-2} and 65×10^{-2}, while all other parameters remain unchanged. Overall, the CPEWS model utilizes approximately 99 M parameters during training, effectively preventing overfitting while maintaining sufficient representational capacity. This balance ensures stable training and robust performance in weakly supervised settings.

The hyperparameter settings in Table 1 were optimized using various strategies. The momentum reference was set according to [34]. The weight decay factor and warm-up learning rate were optimized via cross-validation. Initially, common values (e.g., 0.01, 0.001, 0.0001) were chosen, followed by K-fold cross-validation to evaluate different hyperparameter combinations. Ultimately, the best-performing set of parameters was selected. The number of network iterations was determined based on prior literature [35]. Experimental results indicate that the MIoU value achieves optimal performance when the batch size is set to 4. The parameters for the number of prototypes K and $\mathscr{T}$ were set according to the values specified in [26] and [36], respectively. The loss weight parameters λ_1, λ_2 and λ_3 were initially set based on the relative importance of each loss term in the early stages of the experiment, and then fine-tuned throughout the training process based on model stability and performance. The threshold values α and β were determined through multiple rounds of testing, guided by relevant literature [35] and studies on similar tasks [20].

Table 1: Experimental parameter table

Parameter name	Value
Optimizer	PolyWarmupAdamW
Momentum	9×10^{-1}
Weight decay	1×10^{-2}
Learning rate	1×10^{-6}
Iterations (VOC)	20,000
Batch size	4
α	25×10^{-2}
β (VOC)	7×10^{-1}
K	10
$\mathscr{T}$	13
λ_1	12×10^{-2}
λ_2	15×10^{-5}
λ_3	2×10^{-1}
Iterations (COCO)	80,000
β (COCO)	65×10^{-2}

4.2 Experimental Results

Table 2 presents the comparison of MIoU performance of the CPEWS model on the PASCAL VOC 2012 and MS COCO 2014 datasets. The CPEWS model achieved MIoU reached of 69.8% on the validation set and 72.6% on the test set of the PASCAL VOC 2012 dataset, and 41.4% on the validation set of the MS COCO 2014 dataset. These results surpass those of state-of-the-art single-stage methods, demonstrating the significant performance advantages of CPEWS in processing these datasets.

Table 2: Semantic segmentation results

Methods	Publication	Ext.	Backbone	VOC		COCO
				Val (%)	Test (%)	Val (%)
Multi-stage WSSS methods						
RIB [37]	NeurIPS'2021	$\mathscr{I}+\mathscr{S}$	DL-V2	70.2	70.0	–
EPS [28]	CVPR'2021	$\mathscr{I}+\mathscr{S}$	DL-V2	71.0	71.8	–
L2G [38]	CVPR'2022	$\mathscr{I}+\mathscr{S}$	DL-V2	72.1	71.7	44.2
RCA [23]	CVPR'2022	$\mathscr{I}+\mathscr{S}$	DL-V2	72.2	72.8	36.8
Du et al. [39]	CVPR'2022	$\mathscr{I}+\mathscr{S}$	DL-V2	72.6	73.6	–
RIB [37]	NeurIPS'2021	$\mathscr{I}$	DL-V2	68.3	68.6	43.8
ReCAm [7]	CVPR'2022	$\mathscr{I}$	DL-V2	68.4	68.2	45.0
VWL [40]	IJCV'2022	$\mathscr{I}$	DL-V2	69.2	69.2	36.2
W-OoD [30]	CVPR'2022	$\mathscr{I}$	WR38	70.7	70.1	–
MCTformer [19]	CVPR'2022	$\mathscr{I}$	WR38	71.9	71.6	42.0
ESOL [41]	NeurIPS'2022	$\mathscr{I}$	DL-V2	69.9	69.3	42.6
Single-stage WSSS methods						
MBDA [14]	TIP'2023	$\mathscr{I}+\mathscr{S}$	ResNet101	69.5	70.2	37.8

(Continued)

Table 2 (continued)

Methods	Publication	Ext.	Backbone	VOC		COCO
				Val (%)	Test (%)	Val (%)
RPM [16]	AAAI'2020	$\mathscr{I}$	WR38	62.6	62.9	–
1 Stage [15]	CVPR'2020	$\mathscr{I}$	WR38	62.7	64.3	–
AFA [17]	CVPR'2022	$\mathscr{I}$	MiT-B1	66.0	66.3	38.9
SLRNet [42]	IJCV'2022	$\mathscr{I}$	WR38	67.2	67.6	35.0
TSCD [43]	AAAI'2023	$\mathscr{I}$	MiT-B1	67.3	67.5	40.1
ToCo [34]	CVPR'2023	$\mathscr{I}$	DeiT-B	69.8	70.5	41.3
CPEWS	–	$\mathscr{I}$	ViT-B	**69.8**	**72.6**	**41.4**

Note: Bold and italic formatting is used for emphasis. Ext. denotes the supervision type. $\mathscr{I}$: Image-level labels; $\mathscr{S}$: Saliency maps.

Performance on VOC validation set. The CPEWS model achieved an MIoU of 69.8%, outperforming advanced single-stage methods by a significant margin—3.8% higher than AFA [17] and 2.6% higher than SLRNet [42]. This result demonstrates that CPEWS significantly enhances the performance of weakly supervised semantic segmentation by effectively integrating contextual information and utilizing intermediate layer-supervised boundary features.

Performance on the VOC test set. The MIoU after CRF post-processing is 72.6%, which is 2.1% higher than the state-of-the-art ToCo [34] method (pre-trained with ImageNet-1k weights). This indicates that the CPEWS model excels in refining segmentation boundaries and handling complex scenes. Notably, even without CRF post-processing, the CPEWS model achieved an MIoU of 71.4%, 1.2% higher than ToCo. This demonstrates that CPEWS not only performs well after post-processing, but also demonstrates strong performance without additional optimization.

Performance on COCO validation set. The CPEWS model achieved an MIoU of 41.4%, outperforming advanced single-stage methods by 2.5% over AFA [17], 6.4% over SLRNet [42], and 0.1% over ToCo [34]. Additionally, the MBDA [14] is 3.6% higher than that of recent single-stage methods utilizing additional supervision. These results demonstrate that the CPEWS model significantly enhances the performance of weakly supervised semantic segmentation, setting a new state-of-the-art for single-stage methods.

4.3 Ablation Study

To investigate the impact of key modules on the model, extensive ablation studies were conducted on the PASCAL VOC 2012 dataset to validate their effectiveness. Specifically, each key module was gradually removed from the model to observe the resulting changes in performance. This approach clarifies the role and contribution of each module within the overall model. Additionally, *t*-tests were conducted on two models, Data-efficient Image Transformers (DeiT) [44] and ViT-B, to validate the effectiveness of the models.

The experimental results in Table 3 indicate that the number of prototype updates significantly impacts model performance. The experiments involved 0, 1, 2, and 5 prototype updates. Without prototype updates, the MIoU achieves 35.6%. With five updates, the MIoU reached 45.5%. However, due to the high resource consumption required for prototype computation, the number of updates was gradually reduced. With two updates, the MIoU increased to 61.5%. As the number of updates decreased, the MIoU showed a gradual upward trend. When the prototype was calculated before training and not updated during the training process, the MIoU reached 64.5%.

Table 3: Ablation studies of prototype

pro_0	pro_1	pro_2	pro_5	MIoU (%)
✓	–	–	–	35.6
–	–	–	✓	45.5
–	–	✓	–	61.5
–	✓	–	–	64.5

Note: pro_0: Without prototype updates. pro_1: prototype updated once. pro_2: prototype updated twice. pro_5: prototype updated five times.

The experimental results presented in Table 4 highlight the significant impact of the loss function on the model's segmentation performance. When using only $\mathscr{L}_{ISM}$, the model achieved an M^h MIoU of 65.0%, an M^m MIoU of 63.0%, and a Seg MIoU of 61.0%. To improve the distinction between the foreground and co-occurring background, $\mathscr{L}_{PDL}$ was introduced, resulting in an MIoU of 67.0%, an MIoU of 65.4%, and a seg MIoU of 64.4%. Although $\mathscr{L}_{PDL}$ effectively separates foreground from background, the model cannot optimize parameters using the provided information. To address this, $\mathscr{L}_{SSL}$ was introduced, leading to an improvement in M^h MIoU to 72.7%, M^m MIoU to 68.3%, and Seg MIoU to 69.8%.

Table 4: Ablation studies of main loss function

$\mathscr{L}_{ISM}$	$\mathscr{L}_{PDL}$	$\mathscr{L}_{SSL}$	M^h (%)	M^m (%)	Seg (%)
✓	–	–	65.0	63.0	61.0
✓	✓	–	67.0	65.4	64.4
✓	✓	✓	72.7	68.3	69.8

Note: M^h: CAM from the final layer. M^m: CAM from the intermediate layer. Seg: semantic segmentation results.

The experimental results in Table 5 demonstrate the impact of the number of Transformer blocks on the model's segmentation performance. Shallow blocks in ViT fail to capture high-level semantic information, while deeper blocks can introduce boundary ambiguity. Experiments were conducted with 8, 10, and 12 blocks. The results show that with 8 blocks, M^h MIoU is 64.6%, M^m MIoU is 61.1%, and Seg MIoU is 62.3%. Increasing the blocks to 10, M^h MIoU improves to 67.4%, M^m MIoU slightly decreases to 60.5%, and Seg MIoU rises to 65.0%. With 12 blocks, M^h MIoU further increases to 70.5%, M^m MIoU to 63.0%, and Seg MIoU to 67.2%.

Table 5: Impact of Transformer blocks

Block	M^h (%)	M^h (%)	Seg (%)
#8	64.6	61.1	62.3
#10	67.4	60.5	65.0
#12	70.5	63.0	67.2

Fig. 3 illustrates the variation in MIoU for the DeiT and ViT-B models. A t-test is conducted to compare the MIoU values between the two models, assessing whether the differences are statistically significant and providing a basis for model selection. The analysis shows that while both DeiT and ViT-B exhibit a similar upward trend in MIoU during the initial training stages, DeiT demonstrates significantly lower MIoU in the

later stages compared to ViT-B. This suggests that, for complex tasks or large-scale datasets, DeiT's lighter design and computational constraints may result in slower feature extraction and training speeds. The *t*-test results further confirm this significant difference in MIoU, supporting the conclusion that ViT-B outperforms DeiT overall. Based on these findings, ViT-B emerges as the more suitable backbone network, due to its faster training speed and superior accuracy.

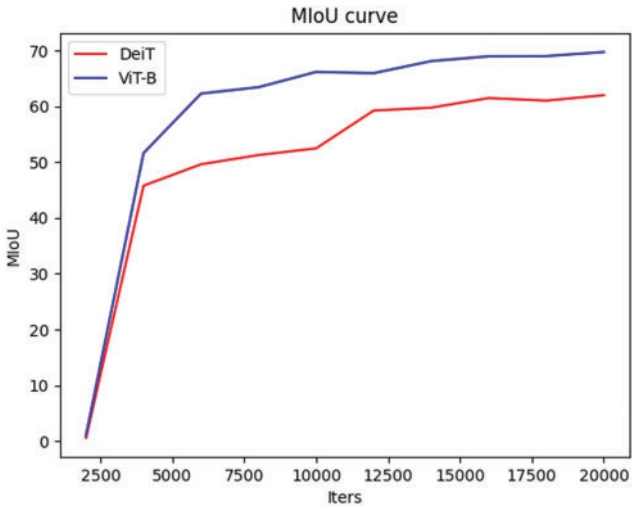

Figure 3: Comparison of the MIoU curves between the DeiT and ViT-B models

4.4 Visualization Results

This section presents and analyzes the visualizations of the CPEWS model on the widely used semantic segmentation datasets PASCAL VOC 2012 and MS COCO 2014. These visualizations not only validate the effectiveness of the CPEWS model but also highlight its advantages in addressing weakly supervised semantic segmentation tasks. The experimental results demonstrate that the CPEWS model excels in resolving issues such as incomplete activation, false activation. By observing the visual results of CAM, pseudo-labels, and semantic segmentation, the paper can clearly illustrate the network's superior performance in tackling these challenges. Additionally, the model's performance during training was monitored and evaluated using loss function and MIoU curves. These figures demonstrate the model's gradual optimization throughout the training process, highlighting how the CPEWS model converges more effectively and enhances overall performance through ongoing refinement and adjustment.

Fig. 4 presents the visualization results of the CAM generated by the CPEWS model. Specifically, Fig. 4a highlights the CPEWS model's significant effectiveness in addressing the problem of incomplete activation. By employing ViT as the backbone network and leveraging the intermediate layer to supervise the final layer features, the network can more effectively activate the target category regions comprehensively. Fig. 4b illustrates the model's effectiveness in mitigating false activation. By learning from contextual prototypes and incorporating $\mathscr{L}_{PDL}$ and $\mathscr{L}_{SSL}$, the network effectively minimizes the mislabeling of background regions as target categories.

Fig. 5 illustrates the effectiveness of the proposed method, which utilizes ViT as the backbone network to capture global information and enhance pseudo label quality. By adding an additional linear classification layer to the network's intermediate features and incorporating $\mathscr{L}_{ISM}$, the generation of the final layer features is effectively supervised. This allows the network to receive supervision not only at the final output

but also at the intermediate layers. Moreover, with the introduction of $\mathscr{L}_{PDL}$ and $\mathscr{L}_{SSL}$, the distinction between the foreground and background is significantly improved. These improvements maintain the semantic diversity of the generated pseudo labels while also increasing their accuracy and clarity by aligning the semantic regions between the intermediate and final layer features. The visualization results in Fig. 5 clearly demonstrate the improvement in pseudo label quality, further validating the effectiveness of the proposed method.

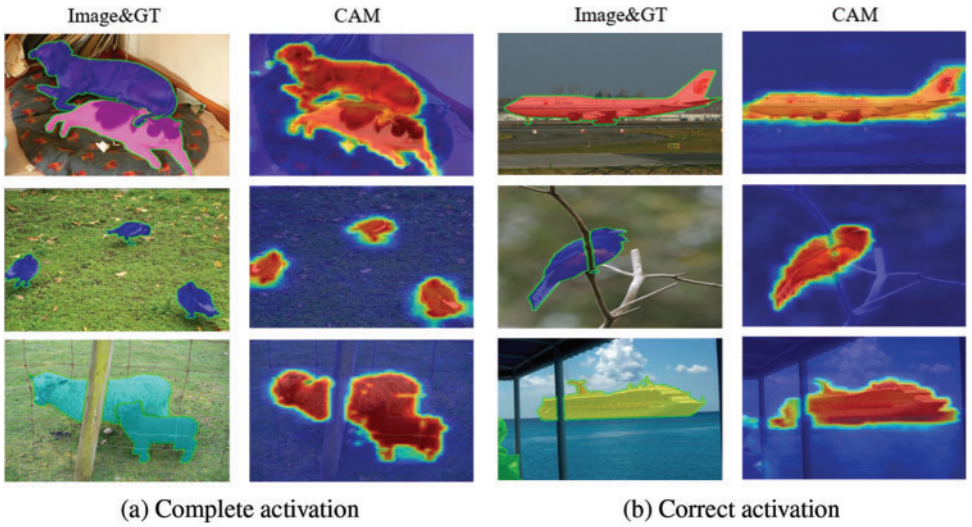

Figure 4: The CPEWS model addresses problems of incomplete activation **(a)** and false activation **(b)** in CAM by introducing CPM, $\mathscr{L}_{PDL}$ and $\mathscr{L}_{SSL}$

Figs. 6 and 7 respectively demonstrate the exceptional performance of the CPEWS model in weakly supervised semantic segmentation tasks on the PASCAL VOC 2012 and MS COCO 2014 datasets. Fig. 6 showcases the semantic segmentation performance of the CPEWS model on the VOC dataset, alongside a comparison with the AFA [17], ToCo [34] methods, and the Ground Truth (GT). Demonstrating the segmentation effects for various categories, such as people, cars, and birds, clearly shows that the CPEWS model generates more accurate segmentation masks for different objects. Fig. 7 presents the semantic segmentation performance of the CPEWS model on the COCO dataset, along with visualizations of CAM and Pseudo Labels. The results demonstrate that the model maintains strong performance even on large-scale datasets.

Fig. 8 shows the loss function variation curve over 20,000 iterations. The graph reveals a steady decrease in the loss function value as training progresses, indicating effective optimization at each iteration and gradual convergence toward the optimal solution. The curve exhibits a smooth decline with a consistent rate of decrease, without any abrupt changes, demonstrating the stability and efficiency of the training process. In the final stages, the loss function stabilized at a low value with no signs of reaching a local minimum, suggesting that the model did not fall into a local optimal solution. This consistent optimization throughout the training process highlights the model's robust performance and successful convergence.

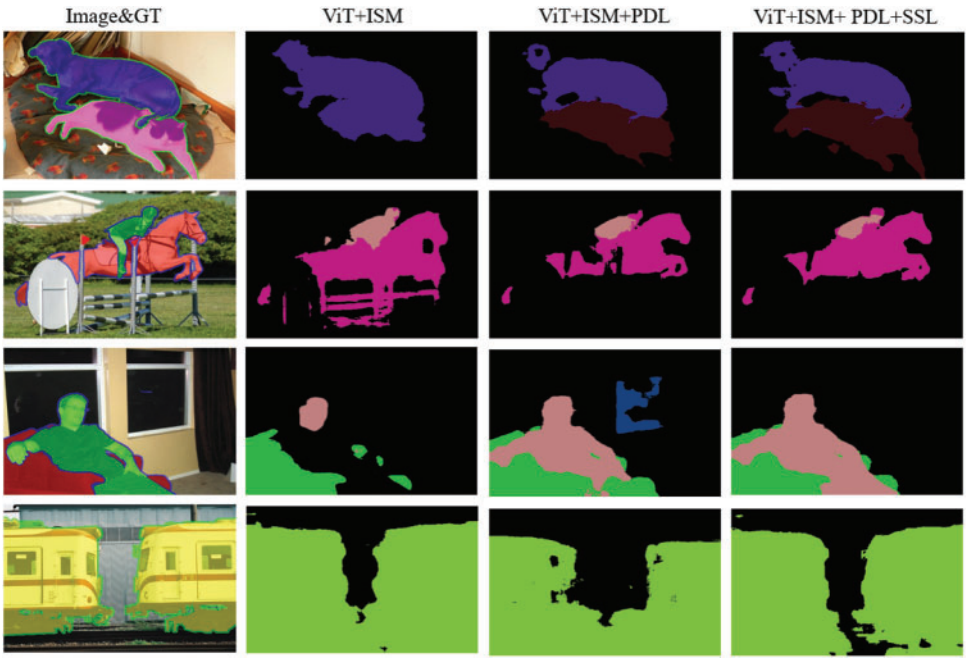

Figure 5: Second column implements an improved ViT with $\mathscr{L}_{ISM}$ to leverage intermediate layer feature knowledge for supervising the final layer features. Third column introduces $\mathscr{L}_{PDL}$ to enhance the separation between foreground and background. Fourth column adds $\mathscr{L}_{SSL}$ to further distinguish the foreground from the co-occurring background

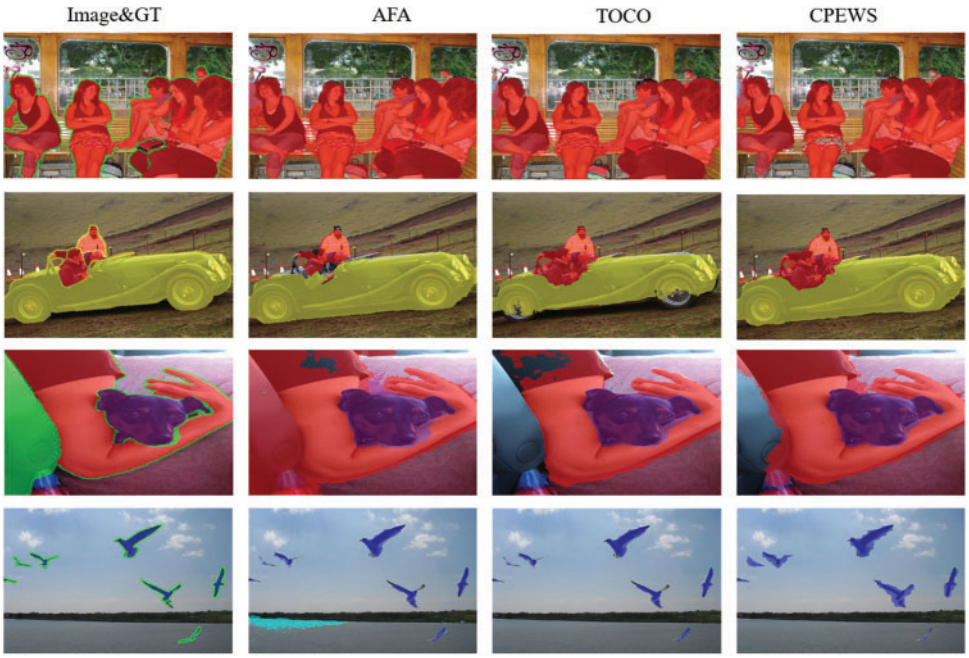

Figure 6: The visualization results of semantic segmentation on the VOC dataset are compared with two mainstream methods. The second column shows AFA, and the third column shows ToCo

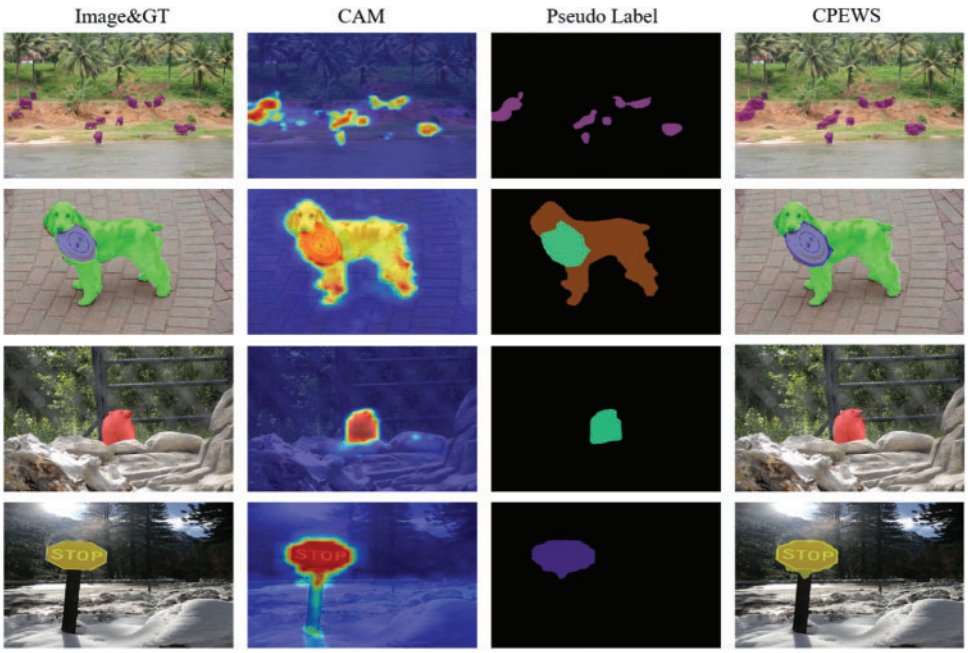

Figure 7: The visualization results of semantic segmentation on the COCO dataset. CAM (second column) and Pseudo Labels (third column)

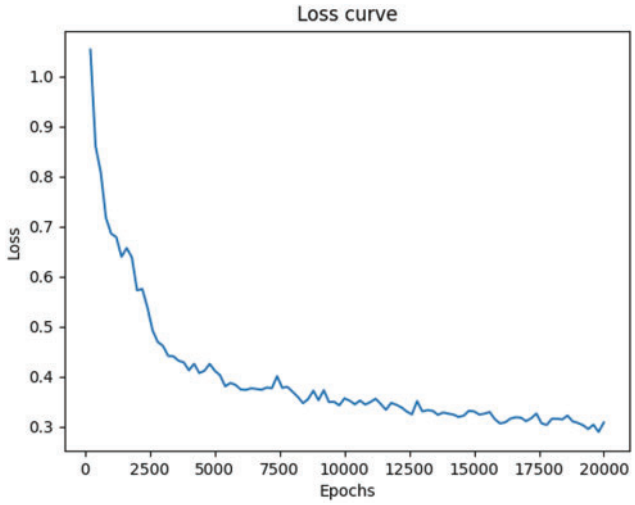

Figure 8: The loss curve indicates that as the number of training iterations increases, the loss function value steadily decreases, following a consistent downward trend

Fig. 9 shows the progression of MIoU as the number of iterations increases. As training advances, the MIoU value steadily rises, indicating that the model is continuously optimizing with each iteration, leading to consistent improvements in segmentation performance. This reflects the model's active learning and adaptation, gradually enhancing its segmentation capability. Moreover, the sustained increase in MIoU suggests that the training process remains stable, with no signs of stagnation or overfitting. The improvement across all categories, rather than just a few, further confirms that the model is effectively learning. The

consistent rise in MIoU further validates the effectiveness of the loss function and optimization strategy, as the model successfully learns and enhances its ability to recognize target categories at each stage of training.

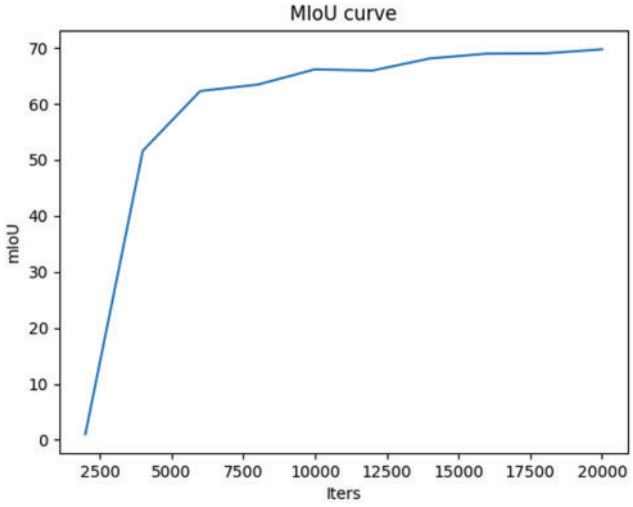

Figure 9: The trend of MIoU as the number of iterations increases

5 Conclusion

This paper introduces a novel end-to-end weakly supervised semantic segmentation model called CPEWS, which leverages ViT as the backbone network for feature extraction and global information capture. Specifically, by utilizing the intermediate feature layers of ViT, which retain rich semantic information, the ISM module is employed to supervise the patch tokens of the final layer. Aligning the semantic regions of the intermediate and final layer features using $\mathscr{L}_{ISM}$, significantly reducing boundary ambiguity and addressing incomplete activation issues. Additionally, the CPM is used to obtain positive and negative sample prototypes for specific classes. Based on these prototypes, $\mathscr{L}_{PDL}$ and $\mathscr{L}_{SSL}$ are proposed to enhance the distinction between foreground and background, effectively mitigating false activation without additional supervision. Extensive experiments conducted on the VOC and COCO datasets demonstrate that the CPEWS model outperforms existing state-of-the-art end-to-end weakly supervised semantic segmentation methods. Although the CPEWS model has achieved the desired accuracy, potential for improvement remains when compared to large-scale supervised learning models. The current parameter count balances computational efficiency and model capability, however, further optimization can be achieved by refining the network architecture or adopting more efficient designs. These changes would help reduce computational overhead, mitigate the risk of overfitting, and improve training efficiency. Looking ahead, achieving a better balance between accuracy and model simplicity can be accomplished by exploring lighter network architectures, pruning techniques, or hybrid precision training methods. These approaches would help optimize the use of computational resources. Additionally, improving training strategies, such as incorporating more effective data augmentation and regularization techniques, would not only boost model accuracy but also further reduce its parameter size. From a broader perspective, this approach offers a new methodology for generating class prototypes by integrating contextual information, ensuring more accurate feature capture.

Acknowledgement: I would like to express my heartfelt gratitude to everyone who contributed to this paper. Their efforts and insights have been invaluable to the success of this work.

Funding Statement: The study has been supported by funding from the following sources: National Natural Science Foundation of China (U1904119); Research Programs of Henan Science and Technology Department (232102210054); Chongqing Natural Science Foundation (CSTB2023NSCQ-MSX0070); Henan Province Key Research and Development Project (231111212000); Aviation Science Foundation (20230001055002); supported by Henan Center for Outstanding Overseas Scientists (GZS2022011).

Author Contributions: The authors contributed to the paper as follows: Led the research efforts and managed the project timeline: Xiaoyan Shao; Designed the research algorithms and drafted the manuscript: Jiaqi Han; Reviewed and revised the manuscript, overseeing the entire project: Lingling Li; Provided algorithmic support: Xuezhuan Zhao; Assisted in manuscript editing: Jingjing Yan. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data presented in this study are available upon request from the corresponding author.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conficts of interest to report regarding the present study.

Nomenclature

CAM	Class Activation Mapping
CLIP	Contrastive Language-Image Pre-Training
CNNs	Convolutional Neural Networks
CPM	Contextual Prototype Module
CRF	Conditional Random Field
DeiT	Data-efficient Image Transformers
ISM	Intermediate Supervised Module
MAP	Mask Average Pooling
MIoU	Mean Intersection over Union
MLP	Multilayer Perceptron
PAR	Pixel-Adaptive Refinement
PDL	Prototype Discrimination Loss
SSL	Superclass Suppression Loss
ViT	Vision Transformer

Appendix A

Table A1 presents all the formulas used in this paper. These formulas are crucial for understanding the core concepts and technical details, and they serve as a reference for the derivations and calculations underpinning the proposed model.

Table A1: Formulas table

Formulas	Formulas number
$Q_i = XW_i^Q, K_i = XW_i^K, V_i = XW_i^V$	(A1)
$head_i = softmax(\frac{Q_i(K_i)^{\mathrm{T}}}{\sqrt{d_k}})V$	(A2)

(Continued)

Table A1 (continued)

Formulas	Formulas number
$head = Concat(head_1, head_2, \ldots, head_h)$	(A3)
$CAM_c = ReLu(\sum_{i=1}^{d} W^{i,c} F^i)$	(A4)
$\mathscr{L}_{ISM} =$	(A5)
$\frac{1}{N^+}\sum_{Y_i=Y_j}(1 - CosSim(F_h^i, F_h^j)) + \frac{1}{N^-}\sum_{Y_i\neq Y_j}(1 - CosSim(F_h^i, F_h^j))$	
$A_c = \frac{\sum_i CAM_c^i \cdot F^i}{\sum_i CAM_c^i}$	(A6)
$\mathscr{L}_{PDL} =$	(A7)
$-\frac{1}{\sum_{c=1}^{C} y_c}\sum_{c=1}^{C} y_c \cdot [log(R_{pos}(A_c, C_{pos}^c)) + log(1 - R_{neg}(A_c, C_{neg}^c))]$	
$R_{pos}(A_c, C_{pos}^c) = exp - \frac{\left\|A_c - C_{pos}^c\right\|_2}{\mathscr{T}}$	(A8)
$R_{neg}(A_c, C_{neg}^c) = exp - \frac{min\left\|A_c - C_{neg}^c\right\|_2}{\mathscr{T}}$	(A9)
$\Phi_c = \{i \mid min(\parallel F^i - P_{neg}^c \parallel_2) < min(\parallel F^i - P_{pos}^c) \parallel_2\}$	(A10)
$\mathscr{L}_{SSL} = \frac{1}{\sum_{c=1}^{C} y_c \cdot \lvert\Phi_c\rvert}\sum_{c=1}^{C} y_c \cdot \sum_{i\in\Phi_c} CAM_c^i$	(A11)
$\Phi_c = \{i \mid min(\parallel F^i - P_{sneg}^c \parallel_2) < min(\parallel F^i - P_{pos}^c) \parallel_2\}$	(A12)
$\mathscr{L} = \mathscr{L}_{cls}^m + \mathscr{L}_{cls}^h + \lambda_1\mathscr{L}_{ISM} + \lambda_2\mathscr{L}_{PDL} + \lambda_3\mathscr{L}_{SSL}$	(A13)
$MIoU = \frac{1}{C+1}\sum_{i=0}^{c} \frac{G\cap P}{G\cup P}$	(A14)

References

1. Lee J, Kim E, Yoon S. Anti-adversarially manipulated attributions for weakly and semi-supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021. p. 4071–80.
2. Akiva P, Dana K. Towards single stage weakly supervised semantic segmentation. arXiv:2106.10309. 2021.
3. Lee J, Yi J, Shin C, Yoon S. BBAM: bounding box attribution map for weakly supervised semantic and instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021. p. 2643–52.
4. Zhang B, Xiao J, Zhao Y. Dynamic feature regularized loss for weakly supervised semantic segmentation. arXiv:2108.01296. 2021.
5. Ru L, Du B, Wu C. Learning visual words for weakly-supervised semantic segmentation. In: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21); 2021. p. 982–8.
6. Zhang B, Xiao J, Jiao J, Wei Y, Zhao Y. Affinity attention graph neural network for weakly supervised semantic segmentation. IEEE Trans Pattern Anal Mach Intell. 2021;44(11):8082–96.
7. Chen Z, Wang T, Wu X, Hua X-S, Zhang H, Sun Q. Class re-activation maps for weakly-supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022. p. 969–78.
8. Yoon S-H, Kwon H, Kim H, Yoon K-J. Class tokens infusion for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2024. p. 3595–605.

9. Fatima M, Attique Khan M, Shaheen S, Albarakati HM, Wang S, Jilani SF, et al. Breast lesion segmentation and classification using u-net saliency estimation and explainable residual convolutional neural network. Fractals. 2024. doi:10.1142/s0218348x24400607.
10. Ullah W, Javed K, Khan MA, Alghayadh FY, Bhatt MW, Naimi ISAl, et al. Efficient identification and classification of apple leaf diseases using lightweight vision transformer (ViT). Discov Sustain. 2024;5(1):116. doi:10.1007/s43621-024-00307-1.
11. Shao F, Luo Y, Chen L, Liu P, Yang Y, Xiao J. Mitigating biased activation in weakly-supervised object localization via counterfactual learning. arXiv:2305.15354. 2023.
12. Everingham M, Eslami SMA, Van Gool L, Williams CKI, Winn J, Zisserman A. The pascal visual object classes challenge: a retrospective. Int J Comput Vis. 2015;111(1):98–136. doi:10.1007/s11263-014-0733-5.
13. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft COCO: common objects in context. In: Computer Vision-ECCV 2014: 13th European Conference; 2014 Sep 6–12; Zurich, Switzerland: Springer. p. 740–55.
14. Chen T, Yao Y, Tang J. Multi-granularity denoising and bidirectional alignment for weakly supervised semantic segmentation. IEEE Trans Image Process. 2023;32:2960–71. doi:10.1109/TIP.2023.3275913.
15. Yang Z, Fu K, Duan M, Qu L, Wang S, Song Z. Separate and conquer: decoupling co-occurrence via decomposition and representation for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2024. p. 3606–15.
16. Yang Z, Meng Y, Fu K, Wang S, Song Z. Tackling ambiguity from perspective of uncertainty inference and affinity diversification for weakly supervised semantic segmentation. arXiv:2404.08195. 2024.
17. Ru L, Zhan Y, Yu B, Du B. Learning affinity from attention: end-to-end weakly-supervised semantic segmentation with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022. p. 16846–55.
18. Sun W, Zhang J, Liu Z, Zhong Y, Barnes N. GETAM: gradient-weighted element-wise transformer attention map for weakly-supervised semantic segmentation. arXiv:2112.02841. 2021.
19. Xu L, Ouyang W, Bennamoun M, Boussaid F, Xu D. Multi-class token transformer for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022. p. 4310–9.
20. Wu Y, Ye X, Yang K, Li J, Li X. DuPL: dual student with trustworthy progressive learning for robust weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2024. p. 3534–43.
21. Zhou T, Wang W, Konukoglu E, Van Gool L. Rethinking semantic segmentation: a prototype view. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022. p. 2582–93.
22. Chen Q, Yang L, Lai J-H, Xie X. Self-supervised image-specific prototype exploration for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022. p. 4288–98.
23. Zhou T, Zhang M, Zhao F, Li J. Regional semantic contrast and aggregation for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022. p. 4299–309.
24. Lang C, Cheng G, Tu B, Han J. Learning what not to segment: a new perspective on few-shot segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022. p. 8057–67.
25. Kayabaşı A, Tüfekci G, Ulusoy İ. Elimination of non-novel segments at multi-scale for few-shot segmentation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV); 2023. p. 2559–67.
26. Tang F, Xu Z, Qu Z, Feng W, Jiang X, Ge Z. Hunting attributes: context prototype-aware learning for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2024. p. 3324–34.
27. Chen Z, Tian Z, Zhu J, Li C, Du S. C-CAM: causal CAM for weakly supervised semantic segmentation on medical image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022. p. 11676–85.

28. Lee S, Lee M, Lee J, Shim H. Railroad is not a train: saliency as pseudo-pixel supervision for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021. p. 5495–505.
29. Xie J, Hou X, Ye K, Shen L. CLIMS: cross language image matching for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022. p. 4483–92.
30. Lee J, Oh SJ, Yun S, Choe J, Kim E, Yoon S. Weakly supervised semantic segmentation using out-of-distribution data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022. p. 16897–906.
31. Zhang B, Yu S, Wei Y, Zhao Y, Xiao J. Frozen clip: a strong backbone for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2024. p. 3796–806.
32. Jang S, Yun J, Kwon J, Lee E, Kim Y. DIAL: dense image-text alignment for weakly supervised semantic segmentation. arXiv:2409.15801. 2024.
33. Ridnik T, Ben-Baruch E, Noy A, Zelnik-Manor L. ImageNet-21k pretraining for the masses. arXiv:2104.10972. 2021.
34. Ru L, Zheng H, Zhan Y, Du B. Token contrast for weakly-supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2023. p. 3093–102.
35. Wu F, He J, Yin Y, Hao Y, Huang G, Cheng L. Masked collaborative contrast for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV); 2024. p. 862–71.
36. Chen L, Lei C, Li R, Li S, Zhang Z, Zhang L. FPR: false positive rectification for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV); 2023. p. 1108–18.
37. Lee J, Choi J, Mok J, Yoon S. Reducing information bottleneck for weakly supervised semantic segmentation. Adv Neural Inform Process Syst. 2021;34:27408–21.
38. Jiang P-T, Yang Y, Hou Q, Wei Y. L2G: a simple local-to-global knowledge transfer framework for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference On Computer Vision and Pattern Recognition (CVPR); 2022. p. 16886–96.
39. Du Y, Fu Z, Liu Q, Wang Y. Weakly supervised semantic segmentation by pixel-to-prototype contrast. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022. p. 4320–9.
40. Ru L, Du B, Zhan Y, Wu C. Weakly-supervised semantic segmentation with visual words learning and hybrid pooling. Int J Comput Vis. 2022;130(4):1127–44. doi:10.1007/s11263-022-01586-9.
41. Li J, Jie Z, Wang X, Wei X, Ma L. Expansion and shrinkage of localization for weakly-supervised semantic segmentation. Adv Neural Inform Process Syst. 2022;35:16 037–51.
42. Pan J, Zhu P, Zhang K, Cao B, Wang Y, Zhang D, et al. Learning self-supervised low-rank network for single-stage weakly and semi-supervised semantic segmentation. Int J Comput Vis. 2022;130(5):1181–95. doi:10.1007/s11263-022-01590-z.
43. Xu R, Wang C, Sun J, Xu S, Meng W, Zhang X. Self correspondence distillation for end-to-end weakly-supervised semantic segmentation. Proc AAAI Conf Artif Intell. 2023;37(3):3045–53. doi:10.1609/aaai.v37i3.25408.
44. Touvron H, Cord M, Douze M, Massa F, Sablayrolles A, Jégou H. Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning (ICML); 2021; PMLR. pp. 10347–57.