Neural Processing Letters (2024) 56:49
https://doi.org/10.1007/s11063-024-11530-2

Category-Aware Saliency Enhance Learning Based on CLIP for Weakly Supervised Salient Object Detection

Yunde Zhang[1] · Zhili Zhang[2] · Tianshan Liu[3] · Jun Kong[1]

Accepted: 8 January 2024

Abstract
Weakly supervised salient object detection (SOD) using image-level category labels has been proposed to reduce the annotation cost of pixel-level labels. However, existing methods mostly train a classification network to generate a class activation map, which suffers from coarse localization and difficult pseudo-label updating. To address these issues, we propose a novel Category-aware Saliency Enhance Learning (CSEL) method based on contrastive vision-language pre-training (CLIP), which can perform image-text classification and pseudo-label updating simultaneously. Our proposed method transforms image-text classification into pixel-text matching and generates a category-aware saliency map, which is evaluated by the classification accuracy. Moreover, CSEL assesses the quality of the category-aware saliency map and the pseudo saliency map, and uses the quality confidence scores as weights to update the pseudo labels. The two maps mutually enhance each other to guide the pseudo saliency map in the correct direction. Our SOD network can be trained jointly under the supervision of the updated pseudo saliency maps. We test our model on various well-known RGB-D and RGB SOD datasets. Our model achieves an S-measure of 87.6% on the RGB-D NLPR dataset and 84.3% on the RGB ECSSD dataset. Additionally, we obtain satisfactory performance on the weakly supervised E-measure, F-measure, and mean absolute error metrics for other datasets. These results demonstrate the effectiveness of our model.

✉ Jun Kong
kongjun@jiangnan.edu.cn

Yunde Zhang
7221905026@stu.jiangnan.edu.cn

Zhili Zhang
528419003@qq.com

Tianshan Liu
tianshan.liu@connect.polyu.hk

[1] Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Jiangnan University, Wuxi 214122, China

[2] School of Computer Science and Technology, Anhui University, Hefei, China

[3] Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong 999077, China

Published online: 16 February 2024

Keywords Weakly supervised · Salient object detection · Category-aware Saliency Enhance Learning · CLIP

1 Introduction

Salient object detection (SOD) [1, 2] mimics the human attention mechanism to isolate visually prominent objects in a scene. It has been utilized in semantic segmentation [3], image compression [4], object tracking [5], video coding [6], etc.

Most SOD methods follow the fully supervised paradigm, which heavily relies on large datasets of pixel-wise annotations. To reduce the cost of manual labeling, weakly supervised methods that use sparse labels (e.g., image-level category labels [7, 8], captions [9, 10], scribbles [11], bounding boxes [12], points [13], subitizing [14], etc) have been applied to realize a trade-off between time consumption and performance. Among the weakly supervised signals, the image-level category labels can provide the activation regions related to category labels and help deduce the position of the objects. Therefore, weakly supervised SOD method based on image-level category labels are investigated and named WSOD in the paper.

Existing WSOD methods [9, 15] (Fig. 1a) train a classification network on the ImageNet [16] or Microsoft COCO [17] dataset and then transform the features of the classification network to generate class activation maps (CAMs) [18, 19]. Next, conditional random fields (CRF) [20] is applied to optimize CAMs, and last, a saliency network is trained using the optimized CAMs. However, CAMs are inaccurate as pseudo labels because they might highlight only the most discriminative region instead of the whole object [9].

In contrast to these methods, the recently proposed JSM [21] updates pseudo labels using category predictions. Specifically, RGB images are used to predict the saliency maps and further enhanced by estimated depth information. Next, initial handcrafted pseudo labels and the enhanced saliency maps are weighted combined by category predictions to form the updated pseudo labels. Finally, a saliency network is trained by the updated pseudo labels.

Inspired by JSM but different from JSM, we employ contrastive vision-language pre-training (CLIP) [22] to update the pseudo labels. CLIP establishes a correlation between image and text by contrastive learning from 400 million image-text pairs. When category label is used as text prompt, CLIP that models the relation of image and language can be utilized. The image-text alignment ability of CLIP is used to generate a category-aware saliency map. Meanwhile, the zero-shot classification ability of CLIP is used to evaluate the quality of the pseudo saliency map and category-aware saliency map to further update the pseudo label. Here, pseudo saliency maps are used as pseudo labels. In terms of the process, as shown in Fig. 1b, we first design a CSEL that has both a classification head and a segmentation head. It can segment a category-aware saliency map guided by a weakly supervised signal (the salient category); second, the category-aware saliency map and the pseudo saliency map are combined based on two confidence scores that are evaluated by CSEL classification head to update the pseudo labels; last, a SOD network is trained using the gradually updated pseudo labels. During the inference phase, given an image, we only need to use the SOD network to predict the result.

Note that our method needs to use the category labels of SOD images like JSM. It is different from traditional CAM-based methods, which use classification training samples (ImageNet or COCO) to acquire category information and then use CAM as pseudo labels. Fortunately, the CapS dataset [21] provides the category ground truth of RGB-D SOD training samples. Therefore, RGB-D SOD is taken as an example to demonstrate our method.

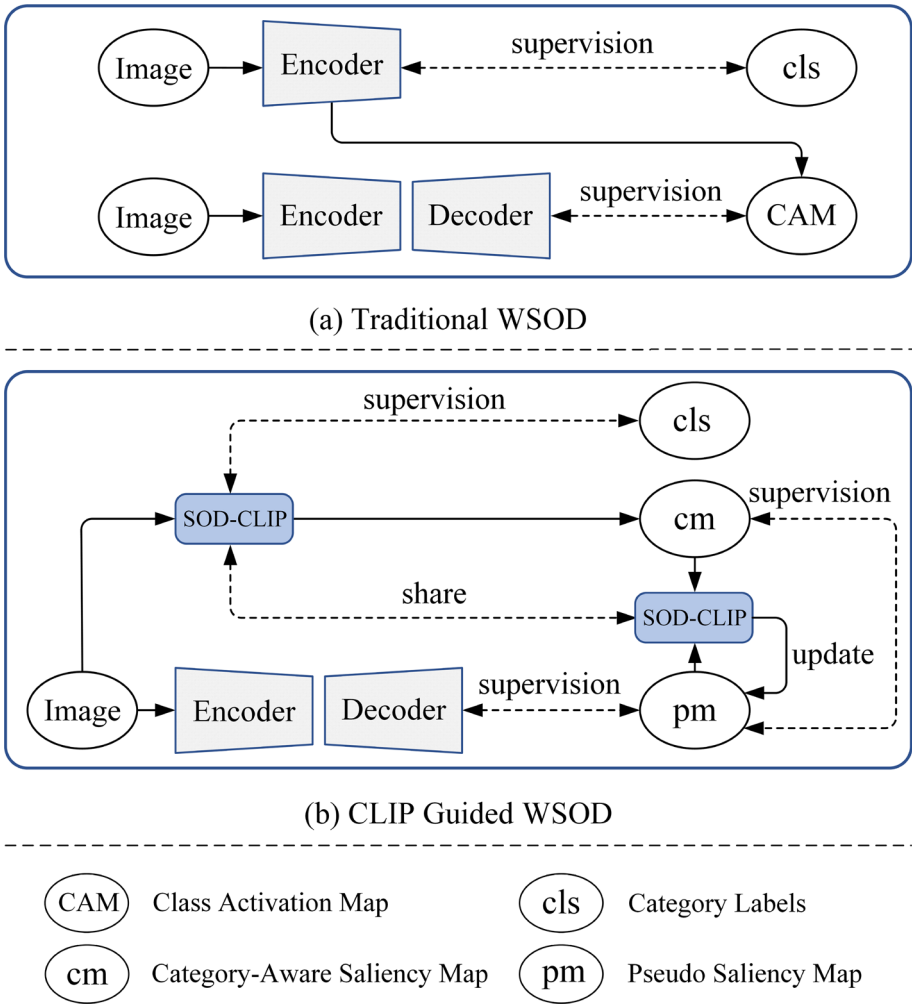

Fig. 1 Comparison between traditional weakly supervised salient object detection methods based on image-level category label (WSOD) and our CLIP guided ones

Meanwhile, to make a fair comparison, color and depth are fused in the input level. More complicated fusion manners are not involved.

The contributions of this paper can be summarized as follows:

- Following the proposal of JSM and the CapS dataset, a weakly supervised SOD method based on category labels is proposed. It outperforms JSM and provides a baseline for further study on WSOD.
- A CSEL is proposed that maintains the zero-shot classification ability of CLIP and builds a bridge from image-level classification to pixel-level saliency prediction.
- CSEL is used to guide the update of the pseudo label. A category-aware saliency map is generated by the CSEL segmentation head under the guidance of salient category text. In addition, the pseudo saliency map is updated by the category-aware saliency map based on their quality evaluations by using the CSEL classification head.

- Experimental results demonstrate that the proposed model achieves satisfactory performance. The win/tie/loss is 13/2/5 and 12/1/7 on RGB-D and RGB SOD dataset compared with existing WSOD methods, respectively.

2 Related Work

2.1 Weakly Supervised Salient Object Detection

Salient object detection has aroused increasing interest. Most methods follow a full-supervised strategy that relies on pixel-level manual annotations. It is a tedious and time-consuming procedure. Outside the constraints, image-level category labels [7, 8], captions [9, 10], scribbles [23, 24], bounding boxes [12], points [13] and subitizing [14] are employed to supervise the training of the model.

The category label based methods [7] use large-scale image datasets to train a classification task, and then generate the CAM as a pseudo label to train the saliency task. ASMO [8] and SCWSOD [25] perform self-training by using the predicted saliency map as the part of pseudo labels. MSW [9] uses mutual-guided multi-task (multi-label classification and caption generation) to help generate pseudo labels. MFNet [26] utilizes a directive filter strategy with pixel-wise and superpixel-wise pseudo labels. NSALWSS [15] is a noise-aware generative adversarial network that emphasizes salient objects and suppresses the noise on the pseudo labels.

The caption based method [10] uses a caption dataset to train a caption generation task and then transfers language-aligned vision features to generate a language-aware saliency map. The scribble based method [23] uses sparse labels indicating foreground and background to achieve whole object detection for RGB images [27], RGB-D images [1], and remote sensing image [11]. The bounding boxes based method [12] leverages the supervision of bounding boxes to update pseudo labels. The point based method [13] integrates point annotations and edges to update pseudo labels. The subitizing based methods [28, 29] introduce counting information into SOD.

Our research focuses on the category label based method. Due to the strong performance of CLIP, we use it to transfer image-text similarity to pixel-text matching for accurate segmentation, while evaluating the quality of the saliency map by its zero-shot classification ability for pseudo label update.

2.2 Weakly Supervised RGB-D Salient Object Detection

RGB-D salient object detection is an active research area in the computer vision community. To lower the cost of manual annotations, some weakly supervised methods for RGB-D SOD have been proposed. DENet [24] trains an RGB-D saliency model with multi-round scribble annotations by self-supervision. JSM [21] mines semantics from caption mask predictions for weakly supervised RGB-D SOD tasks. Concretely, it uses RGB images to predict a saliency map and then uses the estimated depth to refine the saliency map. Next, the refined saliency map combines the pseudo labels using textual semantic weights to update the pseudo labels. Finally, it trains a saliency network using the updated pseudo label.

Inspired by JSM [21], we adopt the same training strategy, but with a different strategy for updating pseudo labels. Guided by the better classification and segmentation abilities of CSEL, our model achieves better performance.

2.3 CLIP-Based Methods

Contrastive vision-language pre-training (CLIP) [30] uses 400 million image-text pairs collected from the internet to train a model with a contrastive loss, where ground-truth image-text pairs are positive samples and mismatched image-text samples are negative samples. The model includes a visual encoder and a language encoder, and maps the input image and text into a unified representation space. CLIP has the ability to align images with any semantic concepts in an open vocabulary for zero-shot classification. It has been applied in point cloud classification [31], open-vocabulary object detection [32, 33], referring image segmentation [34], cross-modal retrieval [35], 3D object language grounding [36], etc.

Since CLIP only models the relation between the whole image and text description, fine-grained alignments (e.g., region and text, pixel and text) are needed for object detection and semantic segmentation. As a result, CLIPSeg [37] equips CLIP with a transformer-based segmentation decoder for segmentation tasks. zsseg [38] combines CLIP with MaskFormer [39] to classify each class-agnostic mask proposal into a category. RegionCLIP [40] and GLIP [41] capture the fine-grained alignment between regular rectangular image regions and textual concepts by region-text pre-training. FILIP [42] achieves fine-grained cross-modal late interaction between image patches and textual tokens. LSeg [43] and DenseCLIP [44] convert the original image-level semantic correlation in CLIP into pixel-level matching for dense prediction. GroupViT [45] divides images into arbitrarily-shaped and semantically-similar segments by progressive merging in a transformer structure, and then establishes the similarity between segment and class prompt by contrastive pre-training. It extends image-level similarity to pixel-level similarity.

Most existing WSOD methods use CAMs as the pseudo labels. Inspired by DenseCLIP [44], our proposed CSEL uses classification text to obtain pseudo label. Meanwhile, our CSEL classification head is also used to evaluate the quality of the pseudo saliency map pm and the category-aware saliency map cm. Besides performing segmentation like DenseCLIP, our CSEL can also perform classification.

3 Proposed Method

3.1 Motivation and Overview

To reduce the cost of pixel-level annotation, a weakly supervised SOD method based on category label is studied in the paper. Existing methods use class activation map (CAM) generated from category information as the pseudo label to train a SOD model. Since CAM might highlight only the most discriminative region instead of the whole object, CAM is not the best choice and need to be improved. In the paper, we introduce CLIP to generate a better pseudo label.

Specifically, our proposed model includes two weakly coupled networks, as shown in Fig. 2a–c. One is the pseudo label update network, and the other is the weakly supervised SOD network. The result of the former is used to supervise the latter. The two networks are trained simultaneously.

Theoretically, the weakly supervised SOD network can use any SOD model. It is trained under the supervision of the pseudo label. Next, we discuss the pseudo label update network.

To obtain a better pseudo label, inspired by DenseCLIP, we design a similar CSEL that performs both classification and segmentation tasks. CSEL models both the image-text similarity

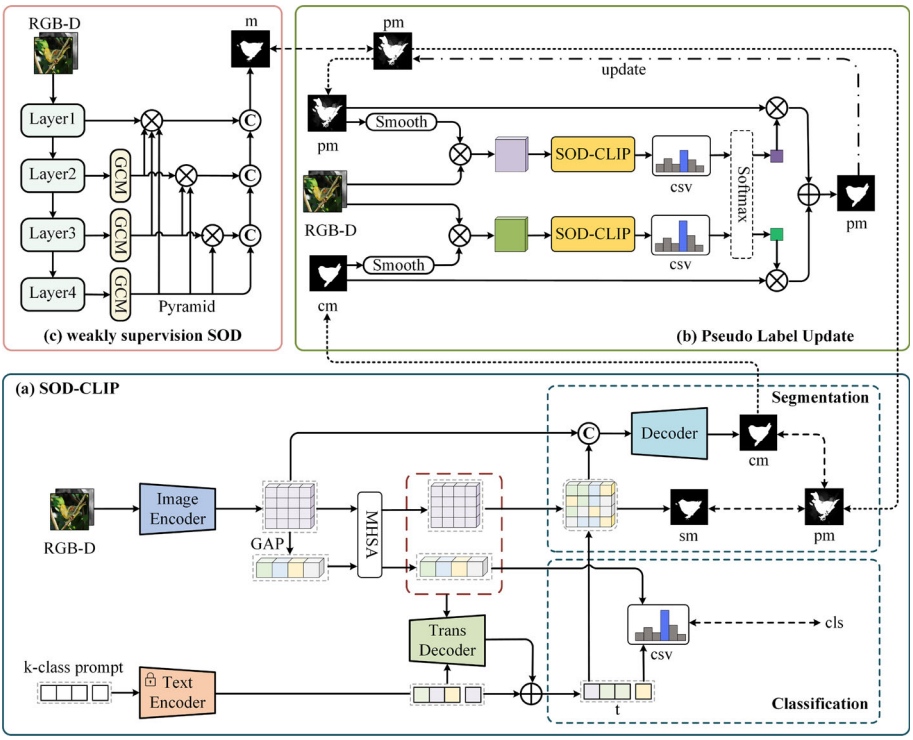

Fig. 2 CLIP guided weakly supervised SOD model

and pixel-text matching relation. In the training stage, the saliency map generated by the traditional handcrafted method is used as the initial pseudo label *pm*. Then, the salient category is used as a weakly supervised signal to train CSEL. It generates a category-aware saliency map *cm* that is further supervised by *pm*. The category-aware saliency map is instinctively better than the initial pseudo saliency map since it aligns well with the language features of the salient category. However, the handcrafted saliency map is also good at perceiving low-level clues. As a result, the pseudo saliency map is updated by combining the pseudo saliency map and category-aware saliency map and using the category-aware confidence scores, which can be inferred from the CSEL classification results. Concretely, the input image is masked by the pseudo saliency map and category-aware saliency map. The higher the quality of the saliency map is, the more accurately the masked input image is classified. Each quality value serves as a confidence score for weighting the pseudo saliency map and category-aware saliency map. By combining the two, the quality of the pseudo saliency map is gradually improved every τ epochs (a training round) during the training process. The category-aware saliency map makes the pseudo saliency map increasingly better, and then the supervision of the better pseudo saliency map generates a better category-aware saliency map. The two maps together become increasingly better to form an optimal supervision signal for the SOD network.

Our method needs category labels of training samples, while SOD training dataset doesn't provide these information. Fortunately, the recently proposed CapS dataset gives the category information of RGB-D SOD training images. Therefore, RGB-D SOD is taken as an example to verify our method. In the paper, simple four-channel fusion is adopted for easy comparison

with existing weakly supervised SOD methods. No complicated fusion of two modalities is involved.

3.2 CSEL

Previous works use the class activation map (CAM) to build a bridge from image-level categories to pixel-level saliency. However, CAMs are not suitable for the pseudo labels of saliency maps, since they might highlight only the most discriminative regions related to categories instead of the whole objects.

The recently proposed CLIP [22] uses 400 million image-text pairs to align the visual and language embedding spaces by contrastive learning. When a set of text prompts about category labels are constructed, CLIP can be regarded as a classifier on an image. Concretely, the similarities between the image and the text prompts are computed, and the class name with the highest score is regarded as the image category. To transfer knowledge from CLIP to downstream dense prediction tasks, DenseCLIP [44] retains the language-compatible feature map besides the global feature, and constructs pixel-text matching relations.

Inspired by this, we use the pipeline of DenseCLIP to construct a **CSEL**. It includes a classification head and a segmentation head. The classification head classifies a given image. The segmentation head segments the category-aware saliency map.

Specifically, input RGB-D image pair $\{r, d\} \in \mathbb{R}^{H\times W\times 4}$ are concatenated and convoluted to obtain a three-channel input $x \in \mathbb{R}^{H\times W\times 3}$.

$$x = Conv(Concat(r, d)) \tag{1}$$

where $Concat(\cdot)$ represents channel concatenation operation and $Conv(\cdot)$ means convolution operation.

CLIP vision encoder encodes x as four-layer features $\{f_i\}_{i=1}^{4} \in \mathbb{R}^{H_i\times W_i\times C_i}$, where H_i and W_i are the height and width of the i-th layer feature, and C_i is the number of channels.

$$\{f_i\}_{i=1}^{4} = \mathcal{E}_I(x) \tag{2}$$

where $\mathcal{E}_I(\cdot)$ is CLIP image encoder.

The feature of the last layer $f_4 \in \mathbb{R}^{H_4\times W_4\times C_4}$ is performed a global average pooling to obtain a global feature $\bar{f_4} \in \mathbb{R}^{1\times C_4}$.

$$\bar{f_4} = GAP(f_4) \tag{3}$$

where $GAP(\cdot)$ denotes global average pooling operation.

Then two are concatenated and performed a multi-head self-attention (MHSA) to generate a category-aware and spatial-sensitive high-layer feature $z \in \mathbb{R}^{H_4\times W_4\times C}$ and a global signal $\bar{z} \in \mathbb{R}^{1\times C}$.

$$[\bar{z}, z] = MHSA([\bar{f_4}, f_4]) \tag{4}$$

where $[\cdot]$ is concatenation operation along patches, and $MHSA(\cdot)$ represents multi-head self-attention layer.

Meanwhile, CLIP text encoder encodes a set of learnable K-class text prompts as text features $t \in \mathbb{R}^{K\times C}$.

$$t = \mathcal{E}_T([p, e_1, e_2, \cdots, e_K]) \tag{5}$$

where $\mathcal{E}_T(\cdot)$ is CLIP text encoder, $p \in \mathbb{R}^{N\times C}$ is learnable textual contexts, N is a hyperparameter specifying the number of context tokens [44], $e_k \in \mathbb{R}^C (1 \le k \le K)$ is the embedding for each category name, K represents the number of salient categories.

To make the text features t more accurate, Transformer decoder [46] is employed to aggregate visual contexts. The text feature t is used as query (Q), and the image feature $[\bar{z}, z]$ is used as key (K) and value (V). The Transformer decoder decodes Q by Q itself, and then by K and V to generate more accurate text features that aligns well with visual clues. Then the text feature is updated through a residual connection.

$$t \leftarrow t + \alpha TransDecoder(t, [\bar{z}, z]) \tag{6}$$

where the learnable parameter α controls the scaling of the residual.

Next, we compute the image-text **class similarity vector** $csv \in \mathbb{R}^{1\times K}$ by using the global signal $\bar{z}$ and the text feature t by:

$$csv = Norm(\bar{z}) \cdot Norm(t)^T \tag{7}$$

where $Norm(\cdot)$ means l_2 normalization operation along the channel, " $\cdot$ " denotes matrix multiplication.

Meanwhile, we compute the pixel-text **matching score matrix** $score \in \mathbb{R}^{H_4\times W_4\times K}$ using the category-aware and spatial-sensitive high-layer feature map z and the text features t by:

$$score = Norm(z) \cdot Norm(t)^T \tag{8}$$

The matching score matrix $score$ is further fed into a convolution layer and a sigmoid activation layer to generate the matching score map sm by:

$$sm = Sigmoid(Conv(score)) \tag{9}$$

where $Sigmoid(\cdot)$ is sigmoid activation function.

Last, the matching score matrix $score$ that explicitly incorporates language priors is concatenated with the feature of the last layer f_4, and then fed into an image decoder to generate **category-aware saliency map** cm.

$$cm = Decoder(Concat(f_4, score)) \tag{10}$$

where $Decoder(\cdot)$ means the decoding process with progressive deconvolution and concatenation.

In the training process, we use salient category cls to supervise class similarity vector csv, and simultaneously use pseudo saliency map pm to supervise sm and cm.

$$loss_T = loss_{ce}(csv, cls) \tag{11}$$

$$loss_I^{high} = loss_{ppa}(sm, pm) \tag{12}$$

$$loss_I^{low} = loss_{ppa}(cm, pm) \tag{13}$$

where $loss_{ce}(\cdot,\cdot)$ is cross entropy loss, and $loss_{ppa}(\cdot,\cdot)$ is pixel position aware loss [47].

The total loss is the average of above losses.

$$loss = (loss_T + loss_I^{high} + loss_I^{low})/3 \tag{14}$$

The proposed CSEL uses image-text alignment ability of CLIP to transfer category clue for the category-aware saliency map. The category-aware and spatial-sensitive feature map z is used to achieve dense prediction, and meanwhile global signal $\bar{z}$ is also utilized to compute the class similarity. It is different from DenseCLIP [44] that uses only z to establish the relation of pixel and category for the fully supervision semantic segmentation.

3.3 Pseudo Label Update

The traditional handcrafted method [48] is employed to generate the initial pseudo label, i.e., pseudo saliency map. It is provided by the CapS dataset. Then, we also obtain the category-aware saliency map from the previous section. The category-aware saliency map has aggregated category knowledge and is intuitively better than the unsupervised handcrafted saliency map. However, the handcrafted saliency map is superior in perceiving low-level clues. Therefore, the category-aware saliency map is employed to gradually update the pseudo saliency map. Since there is no pixel-level ground truth in weakly supervised methods, we cannot evaluate the qualities of the category-aware saliency map and the pseudo saliency map. However, if we consider the two binary saliency maps as masks, the input image with the better mask should be classified into the correct class. Therefore, we use the zero-shot classification ability of CLIP to indirectly evaluate the qualities of the two saliency maps that are regarded as the confidence scores. Based on the confidence scores, the two saliency maps are fused to update the pseudo label.

Specifically, the pseudo label pm and the category-aware saliency map cm are smoothed to serve as masks to make salient part visible and non-salient part unseen in input images. The operation generates pseudo saliency map masked input x_{pm} and category-aware saliency map masked input x_{cm}.

$$x_j = Conv(x \times Smooth(j)) \tag{15}$$

where $j \in \{pm, cm\}$ represents any one of the two masked inputs, $Smooth(\cdot)$ is a Gaussian smooth operation [21], and " $\times$ " is element-wise multiplication.

Then two masked images are fed into CSEL classification part to generate two confidence scores. Note that text encoder of CSEL is frozen.

Concretely, two masked images are fed into image encoder to generate the category-aware global signals $\bar{z}_{pm}, \bar{z}_{cm}$ by:

$$\{f_{i,j}\}_{i=1}^{4} = \mathcal{E}_I(x_j) \tag{16}$$

$$\bar{f}_{4,j} = GAP(f_{4,j}) \tag{17}$$

$$\left[\bar{z}_j, z_j\right] = MHSA\left(\left[\bar{f}_{4,j}, f_{4,j}\right]\right) \tag{18}$$

Note that here we only retain the global signal $\bar{z}$ although category-aware and spatial-sensitive feature z is also generated. That is to say, we only use the classification ability of CSEL.

Next, CSEL serves as a classifier for masked images by computing the similarity between the global signal of masked image and the text embedding of salient categories.

$$c_j = Norm(\bar{z}_j) \cdot Norm(t)^T \tag{19}$$

Then, we take out similarity values for a given category label, and compute softmax to generate the confidence scores.

$$\left[score_{pm}, score_{cm}\right] = softmax(\left[c_{pm}[cls], c_{cm}[cls]\right]) \tag{20}$$

where $softmax(\cdot)$ means softmax function.

Last, two saliency maps are weighted and summed using each confidence score followed by a post-processing by:

$$pm \leftarrow CRF(score_{pm} \times pm + score_{cm} \times cm) \tag{21}$$

where $CRF(\cdot)$ is a fully-connected conditional random field operation [20].

Algorithm 1 Pseudo codes for the update network of pseudo labels.

Input: training images $D = \{r^n, d^n\}_{n=1}^N$; initial pseudo label pm; salient category label cls; the granularity of updating τ;
Output: updated pseudo label pm;
/* training stage */;
for every epoch **do**
  for $\{r, d\} \in D$ **do**
    $x = Conv(Concat(r, d))$;
    $\{f_i\}_{i=1}^4 = \mathcal{E}_I(x)$; //image CLIP
    $\bar{f}_4 = GAP(f_4)$;
    $[\bar{z}, z] = MHSA([\bar{f}_4, f_4])$;
    $t = \mathcal{E}_T([p, e_1, e_2, \cdots, e_K])$;//text CLIP
    $t \leftarrow t$+α TransDecoder(t,[$\bar{z}$,z]);
    $csv = Norm(\bar{z}) \cdot Norm(t)^T$;
    $score = Norm(z) \cdot Norm(t)^T$;
    $sm = Sigmoid(Conv(score))$;
    $cm = Decoder(Concat(f_4, score))$;
    $loss_T = loss_{ce}(csv, cls)$;
    $loss_I^{high} = loss_{ppa}(sm, pm)$;
    $loss_I^{low} = loss_{ppa}(cm, pm)$;
    $loss = (loss_T + loss_I^{high} + loss_I^{low})/3$;
    /* update stage */;
    if epoch$\%\tau == 0$ **then**
      for $j \in \{pm, cm\}$ **do**
        $x_j = Conv(x \times Smooth(j))$;
        $\{f_{i,j}\}_{i=1}^4 = \mathcal{E}_I(x_j)$;
        $\bar{f}_{4,j} = GAP(f_{4,j})$;
        $[\bar{z}_j, z_j] = MHSA([\bar{f}_{4,j}, f_{4,j}])$;
        $c_j = Norm(\bar{z}_j) \cdot Norm(t)^T$;
      end for
      $[score_{pm}, score_{cm}]$
      $= softmax([c_{pm}[cls], c_{cm}[cls]])$;
      $pm \leftarrow CRF(score_{pm} \times pm + score_{cm} \times cm)$;
    end if
  end for
end for
return pm.

At the end of each training round (τ epochs), our update strategy is performed to generate an up-to-date pseudo saliency map.

The algorithm of pseudo label update network is described in Algorithm 1.

3.4 Weakly Supervised SOD

Theoretically, any salient object detection network can be adopted. It is supervised by gradually updating pseudo label described in the previous section. Next, we elaborate our salient object detection network.

Segformer [49] is used as the encoder to obtain four-layer feature $\{G_i\}_{i=1}^4$.

$$\{G_i\}_{i=1}^4 = Segformer(x) \tag{22}$$

Next, global contextual modules (GCMs) [50] are applied in the last three layers to enlarge the receptive field.

$$G_i' = \begin{cases} G_i, & i = 1 \\ GCM(G_i), & i = 2, 3, 4 \end{cases} \tag{23}$$

Further, to improve the representational ability of features, a pyramid multiplication strategy [50] is imposed to enhance low-layer feature by all the other high-layer features.

$$G_i'' = G_i' \times \prod_{k=i+1}^{4} Up(G_k') \tag{24}$$

where $Up(\cdot)$ is upsampling operation.

Next, the decoding process adopts successive upsampling, concatenation and convolution operation.

$$P_i = \begin{cases} BConv(Concat(Up(P_{i+1}), G_i''), & i = 1, 2, 3 \\ G_i'', & i = 4 \end{cases} \tag{25}$$

where $BConv(\cdot)$ is convolution operation with 3×3 kernel followed by a batch normalization layer and a ReLU activation function.

At last, P_1 is performed a convolution and a sigmoid operation to generate the predicted saliency map m that is supervised by the pseudo label pm.

$$m = Sigmoid(Conv(P_1)) \tag{26}$$

The loss of the weakly supervised SOD network is defined as:

$$loss_{saliency} = loss_{ppa}(m, pm) \tag{27}$$

4 Experiments

4.1 Datasets and Evaluation Metrics

Training: Since RGB SOD dataset provides no category information, we use RGB-D SOD dataset to conduct the experiments. In general, most RGB-D models are trained using 1485 images in NJU2K [51] training set and 700 images in NLPR [52] training set, typical works including BBS-Net [50], D3Net [53], MAD [54] and HiDAnet [55]. Some papers will add another 800 DUT [56] training data sets for training, such as works [57, 58]. The purpose of training with two common training sets is to make a fair comparison with existing algorithms. In weak supervision RGB-D SOD, original annotated pixel-level supervision signals can not be used. CapS dataset provides the useful information of RGB-D SOD training samples, such as salient categories, handcrafted unsupervised saliency map [48], caption and mask caption, etc. We use the former two as supervision signal of training and initial pseudo label, respectively.

Testing: In the RGB-D SOD comparison experiments, NJU2K testing dataset (500 samples), NLPR testing dataset (300 samples), STERE [59] dataset (1,000 samples) and DUT [56] testing dataset (400 samples) are tested. In the RGB SOD comparison experiments, ECSSD [60] dataset (1,000 samples), DUTS-Test [7] dataset (5,019 samples), HKU-IS [61] dataset (4,447 samples), DUT-OMRON [62] dataset (5,168 samples), and PASCAL-S [63] dataset (850 samples) are tested. The use of a unified data set for training and testing can

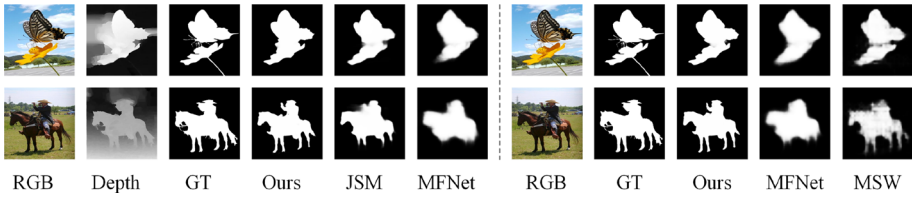

Fig. 3 Visual comparisons with the other state-of-the-art models on RGB-D SOD dataset. The left: the results trained and tested on RGB-D SOD dataset. The right: the results trained and tested on RGB image samples of RGB-D SOD dataset

ensure the results of the experiment and mitigate the performance degradation caused by data differences.

Evaluation: We adopt four widely used metrics to evaluate the performance of our model, including S-measure(S_α) [64], E-measure (E_ξ) [65], F-measure(F_β) [66] and mean absolute error (M) [67].

4.2 Implementation Details

In the training stage, Fig. 2a CSEL and (c) weakly supervised SOD network are simultaneously trained every epoch. Figure 2b pseudo label update is conducted every τ epochs (a train round). In the CSEL, the image encoder adopts the CLIP-ResNet version [22], and the text encoder uses a transformer [46] modified by [68]. In the test stage, Fig. 2 (c) weakly supervised SOD network is only needed.

During the training and testing phases, the input RGB and depth images are resized to 352×352. Multiple enhancement strategies are used for all training images, i.e., random flipping, rotation and border clipping. The parameters of CSEL are initialized with the pretrained parameters of CLIP, where the text encoder of CSEL is fixed to preserve the text knowledge from CLIP. The parameters of the weakly supervised SOD encoder are initialized with the pretrained parameters of Segformer [49]. The remaining parameters are initialized to PyTorch default settings. A training round is set as τ=3 epochs. The Adam optimizer is employed to train our network with a batch size of 10, and the initial learning rate is set to 5e-5. Our model converges within 100 epochs on a NVIDIA GTX 3090 GPU. The model parameters and training time with RGB input and RGB-D input are roughly the same, the parameter is about 190 M, the training time is 22 h and the inference speed is 34 FPS.

4.3 Model Performance

4.3.1 Comparison on RGB-D SOD Dataset

Since RGB-D WSOD was proposed recently, only one comparison method is available, namely, JSM [21]. For a comprehensive comparison, we retrain MFNet [26] with RGB-D four-channel input using their published code. The top part of Table 1 presents a comparison among these two methods and ours trained on the RGB-D SOD training dataset and tested on the RGB-D SOD testing dataset. According to the results, Ours^{RGB-D} performs impressively on most datasets. The visual comparisons in the left part of Fig. 3 also suggest that our

Table 1 Quantitative comparisons on four RGB-D SOD datasets

Methods	Input	FPS	NLPR				NJU2K				STERE				DUT			
			$S_\alpha \uparrow$	$E_\xi \uparrow$	$F_\beta \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\xi \uparrow$	$F_\beta \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\xi \uparrow$	$F_\beta \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\xi \uparrow$	$F_\beta \uparrow$	$M \downarrow$
Ours^{RGB-D}	RGB-D	34	**.876**	**.930**	**.847**	**.034**	.781	.819	**.784**	**.097**	**.853**	**.902**	**.850**	**.060**	**.857**	**.906**	**.867**	**.062**
JSM(NIPS21) [21]	RGB-D	30	.805	.888	.771	.060	.713	.788	.717	.133	.782	.852	.778	.095	.792	.870	.797	.093
MFNet(ICCV21) [26]	RGB-D	15	.821	.849	.701	.063	**.793**	**.845**	.758	.106	.831	.888	.792	.077	.809	.879	.787	.087
Ours$^{RGB/RGB-D}$	RGB	35	**.871**	**.927**	**.850**	**.035**	.765	.811	.766	.103	**.846**	**.899**	**.845**	**.062**	**.843**	**.896**	**.845**	**.068**
MFNet(ICCV21) [26]	RGB	16	.820	.840	.687	.064	**.803**	**.848**	.765	**.102**	.841	.886	.794	.074	.812	.880	.788	.087
MSW(CVPR19) [9]	RGB	37	.826	.841	.697	.076	.784	.841	**.773**	.120	.838	.879	.805	.090	.825	.875	.793	.105

The first three methods denote the model trained and tested on the RGB-D SOD dataset. The last three methods denote the model trained and tested on RGB image samples of RGB-D SOD dataset. bold indicates the best result

Table 2 Quantitative comparisons on five RGB SOD datasets

Methods	ECSSD				DUTS-Test				HKU-IS				DUT-OMRON				PASCAL-S			
	$S\uparrow$	$E_\xi\uparrow$	$F_\beta\uparrow$	$M\downarrow$	$S\uparrow$	$E_\xi\uparrow$	$F_\beta\uparrow$	$M\downarrow$	$S\uparrow$	$E_\xi\uparrow$	$F_\beta\uparrow$	$M\downarrow$	$S\uparrow$	$E_\xi\uparrow$	$F_\beta\uparrow$	$M\downarrow$	$S\uparrow$	$E_\xi\uparrow$	$F_\beta\uparrow$	$M\downarrow$
Ours^{RGB}	**.843**	**.885**	**.871**	**.072**	.769	.833	**.731**	.076	**.857**	.918	**.884**	**.050**	**.774**	**.831**	**.707**	**.073**	.766	.808	**.756**	.113
NSALWSS(TMM22) [15]	.834	.884	.856	.077	**.781**	**.849**	.730	**.073**	.854	**.923**	.864	.051	.745	.801	.648	.088	.768	**.822**	**.756**	**.110**
MFNet(ICCV21) [26]	.834	.885	.854	.084	.775	.839	.710	.076	.846	.921	.851	.059	.742	.803	.646	.087	**.770**	.817	.751	.115
MSW(CVPR19) [9]	.827	.884	.840	.096	.759	.814	.684	.091	.818	.895	.814	.084	.756	.763	.609	.109	.768	.790	.713	.133
ASMO(AAAI18) [8]	.802	.853	.797	.110	.697	.772	.614	.116	-	-	-	-	.752	.776	.622	.101	.717	.772	.693	.149
WSS(CVPR17) [7]	.811	.869	.823	.104	.748	.795	.654	.100	.822	.896	.821	.079	.725	.768	.603	.109	.744	.791	.715	.139

Ours^{RGB} denotes the model trained on RGB training samples of RGB-D SOD dataset and tested on RGB SOD testing samples. The others are trained according to their original settings and tested on RGB SOD testing samples. bold indicates the best result

Table 3 Ablation study about different supervision

	Pseudo Label	NLPR				NJU2K				STERE				DUT			
		$S\uparrow$	$E_\xi\uparrow$	$F_\beta\uparrow$	$M\downarrow$	$S\uparrow$	$E_\xi\uparrow$	$F_\beta\uparrow$	$M\downarrow$	$S\uparrow$	$E_\xi\uparrow$	$F_\beta\uparrow$	$M\downarrow$	$S\uparrow$	$E_\xi\uparrow$	$F_\beta\uparrow$	$M\downarrow$
No.1	init-pm	.755	.803	.625	.098	.700	.768	.642	.165	.749	.812	.690	.130	.727	.818	.672	.140
No.2	m	.784	.853	.678	.072	.722	.797	.682	.136	.780	.838	.731	.102	.803	.879	.782	.088
No.3	cm	.824	.856	.715	.060	**.798**	**.842**	.766	.099	.838	.884	.795	.072	.821	.877	.787	.086
No.4	pm+cm	.835	.870	.744	.056	.778	.824	.738	.109	.825	.866	.770	.077	.826	.883	.792	.082
No.5	Ours(w_1·pm+w_2·cm)	**.876**	**.930**	**.847**	**.034**	.781	.819	**.784**	**.097**	**.853**	**.902**	**.850**	**.060**	**.857**	**.906**	**.867**	**.062**

init-pm: initial pseudo saliency map. *m*: prediction saliency map. *cm*: category-aware saliency map. *pm*: pseudo saliency map. w_1 and w_2: weights of *pm* and *cm* by CSEL classification part. bold indicates the best result

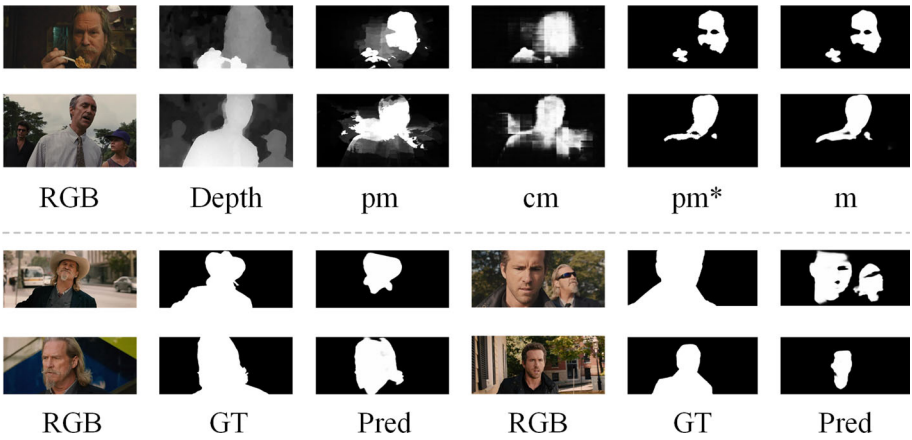

Fig. 4 Some failure cases in NJU2K dataset. The first two rows show two examples of RGB image, depth image, initial pseudo saliency map, category-aware saliency map, updated pseudo saliency map and predicted saliency map of training samples. The last two rows show the predicted results of four similar testing samples. It demonstrates that the SOD network trained by the incomplete updated pseudo labels also generates the wrong predicted results

segmentation accuracy is higher than those of the other methods. The excellent performance is due to the ability of the proposed CSEL to transfer category text clues to segmentation tasks and to accurately classify masked images to obtain confidence scores for updating the pseudo labels. We also find that our method segments objects with different categories. Although only one salient category is assigned to the images in the training set, our method can still find multiple salient objects because the SOD network is weakly related to the salient category prior. An image having only one salient category label has no impact on the performance of our WSOD model.

To further make a fair comparison, we retrain our model using RGB images in RGB-D training image pairs, and compare it with MFNet [26] and MSW [9]. Our contribution to the WSOD task is presented in the bottom part of Table 1 and the right part of Fig. 3. When tested on the RGB image samples of the RGB-D SOD dataset, the proposed language-guided strategy leads to performance improvements.

However, we also find that local bias appears in the NJU2K dataset. By analyzing, we identify the defect. When the initial pseudo saliency map contains an incomplete salient object, the category-aware saliency map supervised by the pseudo saliency map cannot add the missing parts. As a result, updated pseudo saliency map and predicted saliency map also have the same errors. That is, the initial pseudo saliency map has a significant impact on the result. The non-salient part of the initial pseudo saliency map can be gradually suppressed by the category-aware saliency map, but the missing salient part in the initial pseudo saliency map cannot be supplemented. Figure 4 shows some examples. The first two rows present two examples of RGB images, depth images, initial pseudo saliency maps, category-aware saliency maps, updated pseudo saliency maps, and the final predicted saliency maps in the training process. The last two rows show four similar images from the testing dataset. We find that the SOD network trained by the incomplete updated pseudo labels also generates the wrong predicted results.

In terms of the inference speed, as presented in Table 1, our method shows a comparable advantage because our SOD network and pseudo saliency map update network are weakly

coupled. The inference only depends on the SOD network, which is simple and relatively lightweight.

4.3.2 Comparison on RGB SOD Dataset

Since there are no category labels in the RGB SOD dataset, our model cannot be trained on the RGB SOD dataset. However, our RGB-D SOD network and pseudo label update network are weakly coupled. We use RGB training samples in the RGB-D SOD dataset to train our model, and then test it on the RGB SOD testing samples. In our model, Depth is added to the RGB image as an auxiliary fourth dimension, which increases depth information. Comparing the first and fourth rows of Table 1, we find that the input of RGB-D is better than that of RGB on the whole, only the F-measure of NLPR is a little different. Table 2 compares WSS [7], ASMO [8], MSW [9], MFNet [26], NSALWSS [15], and our method. The results show that our method only uses 2,185 training samples to obtain results comparable to those of MFNet and NSALWSS, which uses 10,533 training samples of RGB SOD tasks. We achieve the highest results on all metrics in ECSSD and DUT-OMRON datasets. In all datasets, the metric F-measure also gets the highest result. The win/tie/loss is 13/2/5 and 12/1/7, respectively. We will strive to improve our work to achieve even better performance.

4.4 Ablation Studies About Different Supervision Signals

4.4.1 Quantitative Analysis

We conduct ablation studies on the NLPR, NJU2K, STERE and DUT datasets to investigate the performances of the models under different supervision signals, as presented in Table 3. Note that all the pseudo labels all use CRF post processing. CRF can improve the performance by 0.02 in MAE evaluation metric. In the $1st$ row, when the initial handcrafted pseudo saliency map is used to supervise the training of the model, the model performance is worse. In the $2nd$ row, the supervision signal is updated by the predicted saliency map generated from the weakly supervised SOD network at the end of each training round. The model performance under the self-updating strategy is obviously improved due to the progressively improved supervision signal. In the $3rd$ row, when category-aware saliency map serves as a supervision signal, the model performance using the pixel-text matching ability of CSEL is significantly enhanced. In the $4th$ row, when the category-aware saliency map serves as the residual of the pseudo saliency map, the model performance is comparable to that of the previous model. In the $5th$ row, the weighted fusion of the category-aware saliency map and pseudo saliency map under the guidance of the category classification results achieves the best performance.

The comparison between $1th$, $2nd$ rows and the other rows demonstrates that if there is no our designed CSEL and the corresponding update strategy, SOD model can't achieve a satisfying performance although it uses transformer-based backbone. Furthermore, the comparison between the $4th$ row and the $5th$ row demonstrates that CSEL achieves more accurate classification on masked input images and generates the correct update weights.

4.4.2 Visual Analysis

We also give update process visualization of different maps in Fig. 5. From the figure, we can conclude some conclusions.

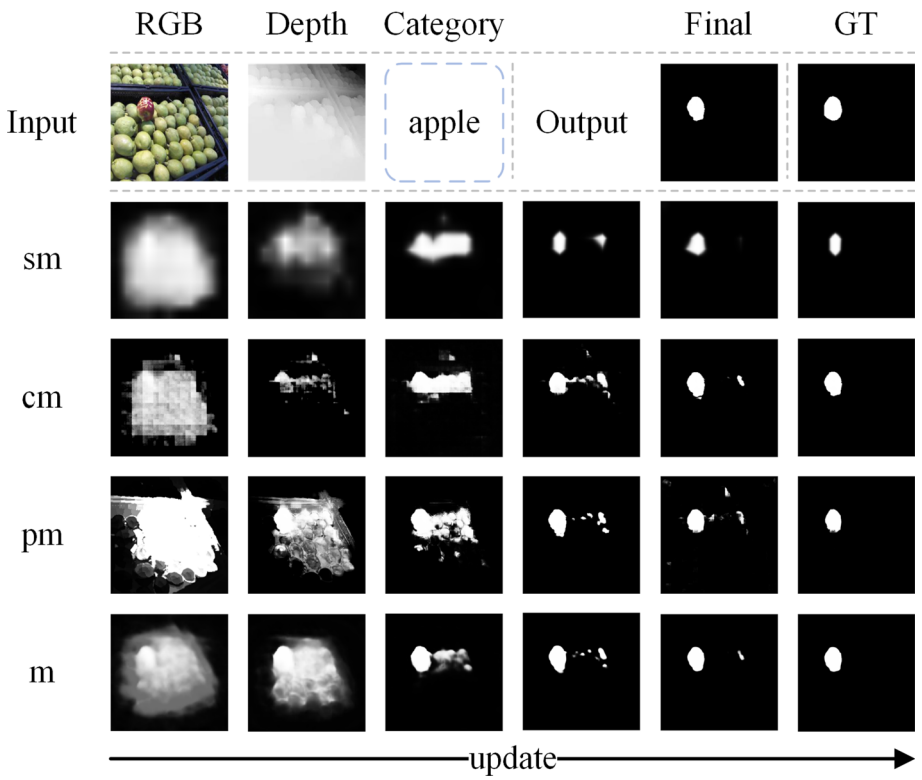

Fig. 5 Update visualization of different maps. from top to bottom: Input & Output & GT, matching score map (sm), category-aware saliency map (cm), pseudo saliency map (pm), predicted saliency map (m). From left to right of the first line: RGB image, depth image, category label, final predicted map, ground truth. From left to right of the other lines: the update process of different maps

- The matching score map *sm* is a low-resolution map, and locates the rough position of salient object.
- In the 1*st* column, the *cm* is category-aware saliency map that is from CSEL. The *pm* is low-quality initial pseudo saliency map. The *cm* is better than *pm* because category prior has been injected.
- The pseudo map *pm* in the next column is the fusion of category-aware map *cm* and pseudo map *pm* from previous column. For example, the *pm* in the 2*nd* column comes from the *cm* and the *pm* in the 1*st* column.
- The pseudo map *pm* is gradually updated to the correct direction.
- The predicted saliency map *m* is nearly the same as pseudo saliency map *pm* because *m* is the output of SOD network supervised by the pseudo saliency map *pm*.
- All of the maps become increasingly better towards the ground truth. The category-aware map *cm* makes pseudo map *pm* the better in the start. Then, the better *cm* is generated by the supervision of the better *pm*. The two together promote each other, making *pm* increasingly better.

5 Conclusions

In this paper, we proposed a weakly supervised SOD method based on image-level category labels. By borrowing the image-text alignment and zero-shot classification abilities of CLIP, we designed a CSEL that consists of a segmentation head and a classification head. It transfers image-text classification to pixel-text matching and generates a category-aware saliency map. Furthermore, it evaluates the quality of the category-aware saliency map and pseudo saliency map and uses quality confidence scores as weights to update the pseudo labels. Finally, the updated pseudo label is used to supervise the SOD network. Experimental results demonstrate the superiority and effectiveness of our method. In the future, we will consider the better utilization of depth images instead of input fusion, and further exploit the language influence of caption mask prediction on the WSOD task.

Acknowledgements This work was partially supported by the National Natural Science Foundation of China (62371209, 62371208), Scientific and Technological Aid Program of Xinjiang (2017E0279), 111 Projects under Grant B12018.

Declarations

Conflict of interest We declare that there is no conflict of interests regarding the publication of this paper.

References

1. Zhao Z, Huang Z, Chai X, Wang J (2023) Depth enhanced cross-modal cascaded network for rgb-d salient object detection. Neural Process Lett 55(1):361–384
2. Wang A, Wang M, Li X, Mi Z, Zhou H (2017) A two-stage bayesian integration framework for salient object detection on light field. Neural Process Lett 46:1083–1094
3. Chen T, Yao Y, Zhang L, Wang Q, Xie G, Shen F (2022) Saliency guided inter-and intra-class relation constraints for weakly supervised semantic segmentation. IEEE Trans Multimed 25:1727–1737
4. Patel Y, Appalaraju S, Manmatha R (2021) Saliency driven perceptual image compression. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision, pp 227–236
5. Zhou Z, Pei W, Li X, Wang H, Zheng F, He Z (2021) Saliency-associated object tracking. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 9866–9875
6. Fischer K, Fleckenstein F, Herglotz C, Kaup A (2021) Saliency-driven versatile video coding for neural object detection. In: ICASSP 2021-2021 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 1505–1509
7. Wang L, Lu H, Wang Y, Feng M, Wang D, Yin B, Ruan X (2017) Learning to detect salient objects with image-level supervision. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 136–145
8. Li G, Xie Y, Lin L (2018) Weakly supervised salient object detection using image labels. In: Proceedings of the AAAI conference on artificial intelligence, pp 7024–7031
9. Zeng Y, Zhuge Y, Lu H, Zhang L, Qian M, Yu Y (2019) Multi-source weak supervision for saliency detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 6074–6083
10. Qian M, Qi J, Zhang L, Feng M, Lu H (2019) Language-aware weak supervision for salient object detection. Pattern Recogn 96:106955

11. Huang Z, Xiang T-Z, Chen H-X, Dai H (2022) Scribble-based boundary-aware network for weakly supervised salient object detection in remote sensing images. arXiv:2202.03501
12. Liu Y, Wang P, Cao Y, Liang Z, Lau RW (2021) Weakly-supervised salient object detection with saliency bounding boxes. IEEE Trans Image Process 30:4423–4435
13. Gao S, Zhang W, Wang Y, Guo Q, Zhang C, He Y, Zhang W (2022) Weakly-supervised salient object detection using point supervison. In: Proceedings of the AAAI conference on artificial intelligence (AAAI), pp 1–9
14. Zheng X, Tan X, Zhou J, Ma L, Lau RW (2021) Weakly-supervised saliency detection via salient object subitizing. IEEE Trans Circuits Syst Video Technol 31(11):4370–4380
15. Piao Y, Wu W, Zhang M, Jiang Y, Lu H (2022) Noise-sensitive adversarial learning for weakly supervised salient object detection. IEEE Trans Multimed 25:2888–2897
16. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L (2009) ImageNet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. IEEE, pp 248–255
17. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft COCO: common objects in context. In: European conference on computer vision. Springer, pp 740–755
18. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2016) Learning deep features for discriminative localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2921–2929
19. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-CAM: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision, pp 618–626
20. Krähenbühl P, Koltun V (2011) Efficient inference in fully connected CRFs with Gaussian edge potentials. Adv Neural Inf Process Syst 24:1–9
21. Li J, Ji W, Bi Q, Yan C, Zhang M, Piao Y, Lu H (2021) Joint semantic mining for weakly supervised RGB-D salient object detection. Adv Neural Inf Process Syst 34:1–15
22. Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, Sastry G, Askell A, Mishkin P, Clark J (2021) Learning transferable visual models from natural language supervision. In: International conference on machine learning. PMLR, pp 8748–8763
23. Yu S, Zhang B, Xiao J, Lim EG (2021) Structure-consistent weakly supervised salient object detection with local saliency coherence. In: Proceedings of the AAAI conference on artificial intelligence (AAAI). AAAI Palo Alto, CA, USA, pp 3234–3242
24. Xu Y, Yu X, Zhang J, Zhu L, Wang D (2022) Weakly supervised RGB-D salient object detection with prediction consistency training and active scribble boosting. IEEE Trans Image Process 31:2148–2161
25. Piao Y, Wang J, Zhang M, Ma Z, Lu H (2021) To be critical: self-calibrated weakly supervised learning for salient object detection. arXiv:2109.01770
26. Piao Y, Wang J, Zhang M, Lu H (2021) MFNet: multi-filter directive network for weakly supervised salient object detection. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 4136–4145
27. Zhou X, Ren Z, Zhou S, Yu T, Jiang Z (2023) Unsupervised saliency detection via knn mechanism and object-biased prior. Neural Process Lett 55:1–15
28. Liu Z, Tan Y, He Q, Xiao Y (2022) Swinnet: swin transformer drives edge-aware rgb-d and rgb-t salient object detection. IEEE Trans Circuits Syst Video Technol 32(7):4486–4497
29. Tian X, Xu K, Yang X, Yin B, Lau RW (2022) Learning to detect instance-level salient objects using complementary image labels. Int J Comput Vis 130(3):729–746
30. Papadopoulos S-I, Koutlis C, Papadopoulos S, Kompatsiaris I (2023) Victor: visual incompatibility detection with transformers and fashion-specific contrastive pre-training. J Vis Commun Image Represent 90:103741
31. Zhang R, Guo Z, Zhang W, Li K, Miao X, Cui B, Qiao Y, Gao P, Li H (2022) PointCLIP: point cloud understanding by CLIP. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 8552–8562
32. Gu X, Lin T-Y, Kuo W, Cui Y (2022) Open-vocabulary object detection via vision and language knowledge distillation. In: International conference on learning representations, pp 1–20
33. Zang Y, Li W, Zhou K, Huang C, Loy CC (2022) Open-vocabulary DETR with conditional matching. arXiv:2203.11876
34. Chen Y (2022) Semantic image segmentation with feature fusion based on Laplacian pyramid. Neural Process Lett 54(5):4153–4170
35. Zeng Z, Mao W (2022) A comprehensive empirical study of vision-language pre-trained model for supervised cross-modal retrieval. arXiv:2201.02772
36. Thomason J, Shridhar M, Bisk Y, Paxton C, Zettlemoyer L (2022) Language grounding with 3D objects. In: Conference on robot learning. PMLR, pp 1691–1701

37. Lüddecke T, Ecker AS (2021) Prompt-based multi-modal image segmentation. arXiv:2112.10003
38. Xu M, Zhang Z, Wei F, Lin Y, Cao Y, Hu H, Bai X (2021) A simple baseline for zero-shot semantic segmentation with pre-trained vision-language model. arXiv:2112.14757
39. Cheng B, Schwing A, Kirillov A (2021) Per-pixel classification is not all you need for semantic segmentation. Adv Neural Inf Process Syst 34:1–12
40. Zhong Y, Yang J, Zhang P, Li C, Codella N, Li LH, Zhou L, Dai X, Yuan L, Li Y (2022) RegionCLIP: region-based language-image pretraining. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 16793–16803
41. Li LH, Zhang P, Zhang H, Yang J, Li C, Zhong Y, Wang, L, Yuan L, Zhang L, Hwang J-N (2022) Grounded language-image pre-training. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10965–10975
42. Yao L, Huang R, Hou L, Lu G, Niu M, Xu H, Liang X, Li Z, Jiang X, Xu C (2022) FILIP: fine-grained interactive language-image pre-training. In: International conference on learning representations, pp 1–21
43. Li B, Weinberger KQ, Belongie S, Koltun V, Ranftl R (2022) Language-driven semantic segmentation. In: International conference on learning representations, pp 1–13
44. Rao Y, Zhao W, Chen G, Tang Y, Zhu Z, Huang G, Zhou J, Lu J (2022) DenseCLIP: language-guided dense prediction with context-aware prompting. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 18082–18091
45. Xu J, De Mello S, Liu S, Byeon W, Breuel T, Kautz J, Wang X (2022) GroupViT: semantic segmentation emerges from text supervision. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 18134–18144
46. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, KaiserŁ, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, pp 5998–6008
47. Wei J, Wang S, Huang Q (2020) F^3Net: fusion, feedback and focus for salient object detection. In: Proceedings of the AAAI conference on artificial intelligence, pp 12321–12328
48. Zhu C, Li G, Wang W, Wang R (2017) An innovative salient object detection using center-dark channel prior. In: Proceedings of the IEEE international conference on computer vision workshops, pp 1509–1515
49. Xie E, Wang W, Yu Z, Anandkumar A, Alvarez JM, Luo P (2021) SegFormer: simple and efficient design for semantic segmentation with transformers. Adv Neural Inf Process Syst 34:1–17
50. Fan D-P, Zhai Y, Borji A, Yang J, Shao L (2020) BBS-Net: RGB-D salient object detection with a bifurcated backbone strategy network. In: European conference on computer vision. Springer, pp 275–292
51. Ju R, Ge L, Geng W, Ren T, Wu G (2014) Depth saliency based on anisotropic center-surround difference. In: 2014 IEEE international conference on image processing (ICIP). IEEE, pp 1115–1119
52. Peng H, Li B, Xiong W, Hu W, Ji R (2014) RGBD salient object detection: a benchmark and algorithms. In: European conference on computer vision. Springer, pp 92–109
53. Fan D-P, Lin Z, Zhang Z, Zhu M, Cheng M-M (2020) Rethinking rgb-d salient object detection: models, data sets, and large-scale benchmarks. IEEE Trans Neural Netw Learn Syst 32(5):2075–2089
54. Song M, Song W, Yang G, Chen C (2022) Improving rgb-d salient object detection via modality-aware decoder. IEEE Trans Image Process 31:6124–6138
55. Wu Z, Allibert G, Meriaudeau F, Ma C, Demonceaux C (2023) Hidanet: Rgb-d salient object detection via hierarchical depth awareness. IEEE Trans Image Process 32:2160–2173
56. Piao Y, Ji W, Li J, Zhang M, Lu H (2019) Depth-induced multi-scale recurrent attention network for saliency detection. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 7254–7263
57. Li C, Cong R, Piao Y, Xu Q, Loy CC (2020) Rgb-d salient object detection with cross-modality modulation and selection. In: Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, proceedings, Part VIII 16. Springer, pp 225–241
58. Zhang M, Yao S, Hu B, Piao Y, Ji W (2023) Dfnet: criss-cross dynamic filter network for rgb-d salient object detection. IEEE Trans Multimed 25:5142–5154
59. Niu Y, Geng Y, Li X, Liu F (2012) Leveraging stereopsis for saliency analysis. In: 2012 IEEE conference on computer vision and pattern recognition. IEEE, pp 454–461
60. Yan Q, Xu L, Shi J, Jia J (2013) Hierarchical saliency detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1155–1162
61. Li G, Yu Y (2015) Visual saliency based on multiscale deep features. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5455–5463
62. Yang C, Zhang L, Lu H, Ruan X, Yang M-H (2013) Saliency detection via graph-based manifold ranking. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3166–3173
63. Li Y, Hou X, Koch C, Rehg JM, Yuille AL (2014) The secrets of salient object segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 280–287

64. Fan D-P, Cheng M-M, Liu Y, Li T, Borji A (2017) Structure-measure: a new way to evaluate foreground maps. In: Proceedings of the IEEE international conference on computer vision, pp 4548–4557
65. Fan D-P, Gong C, Cao Y, Ren B, Cheng M-M, Borji A (2018) Enhanced-alignment measure for binary foreground map evaluation. In: International joint conferences on artificial intelligence organization, pp 698–704
66. Achanta R, Hemami S, Estrada F, Susstrunk S (2009) Frequency-tuned salient region detection. In: 2009 IEEE conference on computer vision and pattern recognition. IEEE, pp 1597–1604
67. Perazzi F, Krähenbühl P, Pritch Y, Hornung A (2012) Saliency filters: contrast based filtering for salient region detection. In: 2012 IEEE conference on computer vision and pattern recognition. IEEE, pp 733–740
68. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I (2019) Language models are unsupervised multitask learners. OpenAI blog 1(8):1–9