SLAM: Semantic Learning based Activation Map for Weakly Supervised Semantic Segmentation

Junliang Chen, Xiaodong Zhao, Minmin Liu, Linlin Shen*
Shenzhen University
{chenjunliang2016,zhaoxiaodong2020,liuminmin2020}@email.szu.edu.cn
llshen@szu.edu.cn

Abstract

Recent mainstream weakly-supervised semantic segmentation (WSSS) approaches based on image-level annotations mainly rely on binary image-level classification with limited representation capacity. In this paper, we propose a novel semantic learning based framework for WSSS, named SLAM (Semantic Learning based Activation Map). We firstly design a semantic encoder to learn semantics of each object category and extract category-specific semantic embeddings from an input image. The semantic embeddings of foreground and background are then integrated to a segmentation network to learn the activation map. Four loss functions, i.e., category-foreground, category-background, activation regularization, and consistency loss are proposed to ensure the correctness, completeness, compactness and consistency of the activation map. Experimental results show that our semantic learning based SLAM achieves much better performance than binary image-level classification based approaches, i.e., around 3% mIoU higher than OC-CSE [13], CPN [37] on PASCAL VOC dataset. Our SLAM also surpasses AMN [16] trained with strong per-pixel constraint and CLIMS [33] utilizing extra multi-modal knowledge. Code will be made available.

1. Introduction

Semantic segmentation is one of the fundamental tasks in computer vision, which aims to assign a category to each pixel of the image. Due to the development of the fully convolutional network (FCN), many fully-supervised semantic segmentation (FSSS) methods [3, 4, 21, 32, 35, 38] have achieved excellent performance and can be widely applied. However, FSSS is built upon the accurate pixel-level segmentation labels, which can be really time consuming and may need huge labour. To reduce the massive resources cost of pixel-level annotation, weakly-supervised semantic segmentation (WSSS) aims at achieving comparable performance with FSSS using weaker supervision, such as bounding boxes [11, 25], scribbles [18, 27], points [2], and image-level labels [17, 26, 29, 37]. These methods follow a two-
stage paradigm: generating pseudo semantic segmentation labels based on these weak labels, and then training an FSSS network using the generated pseudo labels. Among these annotations, image-level labels are the most conveniently acquired ones and have been widely studied. Therefore, in this work, we mainly focus on WSSS with image-level supervision.

As the image-level annotations can not provide accurate location information, most of WSSS methods are built upon Class Activation Map (CAM) [39], which relies on binary image-level classification. The original CAM (shown in the top of Figure 1) is simple and effective, but has an obvious weakness, i.e., under-activation. It only produces high response in the discriminative regions, leading to incompletely activated object regions. To address this issue, recent approaches have attempted to expand the area of CAM [28, 30] or adversarially erase the regions with high response and force the network to include less discriminative regions [13, 17, 29] with a classifier. However, they still suffer from under-activation problem. Besides, they may sometimes falsely activate some background regions. The main reason for the failure of these approaches can be explained by the poor representation capacity of binary image-level classification. On one hand, the binary label can only tell whether a category exists or not in a given image, but fails to provide more details about where and what is the category in the given image. On the other hand, the binary image-level classification tends to build connections with the most relative regions. The most relative regions are mainly the most discriminative object regions of each category, and may include the relative background regions which often co-occur, e.g., the railway in a train image.

In this paper, we propose a novel framework for WSSS based on category semantic, shown in the bottom of Fig. 1. We first propose a semantic encoder to learn the semantics of each category and extract the CSS (Category Specific Semantics) embeddings of input image. The encoder is trained by maximizing the similarity between the CSS embeddings of the image and the category semantics of objects available in the image, at both global and local level. A semantic correlation loss is further imposed to suppress the semantics of different categories available in the image. Based on the semantic encoder and learned category semantics, we further proposed an adversarial training framework to generate the CAM (Classification Activation Map) for each category. Given an input image and the CAM of each category, two complementary images, category-foreground and category-background images can be generated by multiplying the input image with the CAM. Both of the two complementary images will be passed to the semantic encoder to produce the semantic embeddings. While the similarity between the foreground semantics and the category semantics shall be maximized, the similarity between background semantics and the category semantics shall be minimized. Figure 2 shows some examples of semantic embeddings of original image or foreground image, and category semantics. As shown in the figure, there exists clear boundaries between semantic embeddings from different categories. Besides, there exists certain distances between different semantic embeddings from the same category. The above phenomenon shows that the semantic encoder can distinguish the samples from different categories and meanwhile keeps their identity. Experiments show that our SLAM provides more accurate activation maps than the binary image-level classification based approaches, and surpasses some recent approaches in new paradigm.

In this paper, we make the following contributions:

- We propose a novel framework, SLAM (Semantic Learning based Activation Map) for WSSS. Different with image-level classification based approaches, we design a semantic classification based approaches, and surpasses some recent approaches in new paradigm.

- While semantic encoder is trained to maximize the similarity between the embedding of input image with the semantics of object categories available in the image, adversarial losses measuring the similarity of foreground and background semantics with different object categories, are designed to supervise the generation of activation map.

- Experimental results show that our SLAM achieves better performance than image-level classification based approaches, i.e., around 3% mIoU higher than OC-CSE [13], CPN [37] on PASCAL VOC dataset. Our SLAM also surpasses AMN [16] trained with strong per-pixel constraint and CLIMS [33] utilizing extra multi-modal knowledge.

2. Related Work

The prosperity of WSSS benefits a lot from the development of Class Activation Map (CAM) proposed by Zhou et al. [39]. In this section, we introduce different manners to generate CAM, including binary image-level classification and recent approaches in new paradigm.

2.1. Binary Image-level Classification

Most recent methods in WSSS built on CAM are trained with image-level classification constraints. However, there exists an obvious weakness of such image-level classification based approaches, i.e., only the most discriminative regions of the image have high response. This results in an under-activation phenomenon that the segmented regions of CAM are not complete. Many recent methods are dedicated
to overcoming this problem. [9, 12] attempted to expand the initial CAM under the constraint of object boundary. [1, 8, 23] explored the pixel-level relationship to generate high-quality pseudo label using the initial CAM. However, in these methods, the quality of the pseudo label is heavily dependent on the quality of the initial CAM. Therefore, many recent approaches have studied to improve the quality of the initial CAM, e.g., Wang et al. proposed SEAM [28] to refine the CAM via a pixel correlation module which exploits the context of each pixel. However, these above approaches introduced additional complicated modules or training procedures.

Recently, Li et al. proposed a two-stage framework GAIN [17] consisting of CAM generation and classification stages. The two stages share the classifier with the same weights. The second stage minimizes the classification score of the masked image of each category using the same weights. The second stage minimizes the classification stages. The two stages share the classifier with the GAIN [17] consisting of CAM generation and classification network to be trained, and the second stage is pre-trained classifier between the two stages. The first stage is a segmentation network. Kweon et al. [13] improved GAIN [17] by not sharing the weights of the classifier between the two stages. The first stage is a segmentation network to be trained, and the second stage is pre-trained classifier. However, Zhang et al. [37] indicated that the training phase of these methods is unstable as they will still lose some regions more or less, due to the randomness of the hiding process.

Besides, they may still suffer from over-activation problem, i.e. mistaken activation in background regions. It results in very small classification loss in the second stage which exacerbates the training instability.

2.2. Recent New Paradigm

Recently, some approaches explore new paradigms to generate CAM. CLIMS [33] introduces extra text knowledge with Contrastive Language-Image Pre-training (CLIP) [22] to conduct cross language image matching. With the open set knowledge in the CLIP model trained with extra 400 million image to text pairs data, CLIMS can suppress the falsely activated background regions to a certain extent and thus makes the generated CAM more complete. AMN [16] proposes an activation manipulation with a per-pixel classification loss to penalize the most discriminative regions and promote the less discriminative regions using refined initial CAM, and thus generates more complete CAM. MCTFormer [34] introduces advanced transformer architecture to WSSS and achieves much higher performance than convolutional neural network (CNN) based approaches. For fair comparisons, we focus on CNN based approaches in this paper.

3. Method

In this section, we introduce our SLAM framework in details. The overview of SLAM is shown in Fig 3. As shown in the figure, SLAM contains two training stages. In the first stage, we train a semantic encoder with thress losses (global similarity, local similarity and semantic correlation loss) to learn the semantics of each category and extract the CSS of input image. In the second stage, we train the segmentation network based on learned semantics of each category and the pre-trained semantic encoder. To maximize and minimize the similarity between semantics of the same category and different categories, four losses, i.e., category-foreground, category-background, activation regularization and consistency losses, are proposed to ensure that the generated activation maps are correct, complete, compact and consistent.

3.1. Semantic Encoder

3.1.1 Architecture.

The semantic encoder is a CNN-based architecture, designed to learn the semantics of each category and extract the semantic embedding of input image. The semantic encoder consists of a backbone followed by $K$ (the number of categories of the dataset) fully connected (FC) layers. The backbone is a vanilla CNN classifier without the classification layer.

3.1.2 Category-Specific Semantic Representation.

Let $f \in \mathbb{R}^{d \times 1}$ be the last layer with $d$ dimensions of the backbone, the category-specific semantic embedding of the $c$-th category ($c \in \{1, 2, \cdots, K\}$) is calculated as:

$$E_c = ReLU(W_c f),$$

where $W_c \in \mathbb{R}^{d \times d}$ is the weights of the FC layer for the $c$-th category, and $ReLU$ denotes the ReLU function.

We also define a category semantic for each category of the dataset to represent the overall semantics of this category. For category $c$, the category semantic $E_c$ is defined as:

$$E_c = ReLU(W_c),$$

where $W_c \in \mathbb{R}^{d \times 1}$ is the weights of for the $c$-th category of the dataset and randomly initialized.

3.1.3 Loss Functions

In this section, we introduce loss functions for training the semantic encoder. For the semantic encoder, we design three losses (global similarity, local similarity and semantic correlation loss) to learn the semantics of each category and extract the semantic embeddings of input image. We introduce the three losses in details in the following sections.

Global Similarity Loss. Given an input image $I$, the global similarity (GS) loss aims to tell the semantic encoder what categories exist in the image. Let $E = \{E_1, \ldots, E_K\}$
and $\mathcal{E} = \{E_1, \ldots, E_K\}$ be the set of semantic embeddings of the image and category semantics of the dataset, respectively, where $K$ is the number of categories of the dataset. For a specific category $c$, if it exists in the image, we maximize the cosine similarity $\text{Sim}(E_c, E_e)$ between $E_c$ and $E_e$, otherwise we minimize $\text{Sim}(E_c, E_e)$, where $\text{Sim}(x, y)$ denotes the cosine similarity between the input vectors $x$ and $y$:

$$\text{Sim}(x, y) = \frac{x^\top y}{\|x\|_2 \|y\|_2}.$$  

(3)

As $E$ and $\mathcal{E}$ are both passed to ReLU function without any negative values as mentioned above, the value of $\text{Sim}(E_e, E_c)$ is in $[0, 1]$. Let $Y$ be the image-level label of the input image. For category $c$, $Y_c \in \{0, 1\}, Y_c = 1$ denotes that category $c$ exists in the image, otherwise it does not exist in the image. The GS loss $L_{GS}$ is defined as a binary cross entropy (BCE) loss:

$$L_{GS} = -\frac{1}{K} \sum_c Y_c \log(\text{Sim}(E_c, E_e)) + (1 - Y_c) \log(1 - \text{Sim}(E_c, E_e)).$$  

(4)

**Local Similarity Loss.** As global loss focus on the overall semantics, the network might easily focus on the discriminative regions of the object. Hence, we introduce a local similarity (LS) loss to alleviate this problem. The LS loss is designed to promote the similarity between the image-level semantic embeddings and the pixel-level local embedding.

We first build a soft semantic embedding which represents the semantic of the category for each pixel. Let $f_l$ be the local features (here, we choose the layer with stride 4 in the backbone of the semantic encoder as the local features), the local embedding $F$ can be computed as:

$$F = \text{ReLU}(\text{Conv}(f_l, W_1)), \text{ReLU}(\text{Conv}(f_l, W_2)).$$  

(5)

where $W_1$ and $W_2$ are the corresponding weights of the $1 \times 1$ convolutional layers.

For an arbitrary pixel $p = (i, j)$ in the local embedding, we compute a soft attention $A$ for each category $A_c(i, j)$:

$$A_c(i, j) = \frac{\exp(\text{Sim}(F(i, j), E_c))}{\sum_{c'} \exp(\text{Sim}(F(i, j), E_{c'}))}.$$  

(6)

where $c$ denotes the $c$-th category of the dataset.

The soft attention $A$ is then used to generate the soft semantic embedding $S$ for the pixel $p$:

$$S(i, j) = \sum_{c} A_c(i, j) \cdot E_c,$$  

(7)

where $\cdot$ denotes element-wise multiplication.

As the soft semantic embeddings from other images should be dissimilar to the local embedding in the same spatial coordinates, we also minimize the similarity between the soft semantic embeddings and local embeddings from different images. We omit the spatial coordinates $(i, j)$ for convenient illustration. Let $F_p$ and $S_p$ denotes the aligned local embedding and soft-assigned embedding in the same spatial coordinates of image $p$ in a batch, respectively. The LS loss $L_{LS}$ is computed as:

$$L_{LS} = -\frac{1}{B} \log(\text{Sim}(F_p, S_p)) + \sum_{p' \neq p} B \log(\text{Sim}(F_{p'}, S_p)),$$  

(8)

where $B$ denotes the batch size.
Semantic Correlation Loss. While GS and LS ensure the image to contain the semantics of available objects and exclude the semantics of non-existing objects, some objects like train and railways, are always correlated together and can easily be difficult to discriminate. Some special losses shall be designed to help the networks to learn the differences between the semantics of them. Hence, we introduce an semantic correlation (SC) loss to suppress the semantic correlation of the semantic embeddings from different categories of the same image. We minimize the cosine similarity of the semantic embeddings from the different categories of the input image. The SC loss is defined as:

$$
L_{SC} = -\frac{1}{C(C-1)} \sum_{c} \sum_{c' \neq c} \mathbb{I}(Y_c = 1, Y_{c'} = 1)[1 - \log(Sim(E_c, E_{c'}))],
$$

where $C$ is the number of available categories of the image, and $\mathbb{I}(condition) \in \{0, 1\}$ is the indicator function and equal to 1 only when the condition is true.

Overall Loss Function During training, the overall loss function of semantic encoder $L_{enc}$ is defined as:

$$
L_{enc} = L_{GS} + L_{LS} + L_{SC}.
$$

3.2. Segmentation Network

3.2.1 Architecture.

Similar with the CAM-based methods, we use a backbone of a convolutional neuron network (CNN) to train a segmentation network for the generation of the initial activation maps with image-level supervision. The last convolutional layer of the segmentation network generates a set of activation maps. Each activation map is corresponding to a specific category of the dataset. All of the activation maps are normalized to $[0, 1]$ using the sigmoid activation function.

3.2.2 Semantic Learning based Activation Map Generation.

The semantic encoder aims at embedding the category semantic into the segmentation network to make it aware of the semantic of the category while generating activation maps. During the training phase of the segmentation network, the weights of semantic encoder are fixed. As shown in Fig 3, for a given image $I$ and its activation map $M_c$ of category $c$, we construct a complementary image pair for this category, i.e. category-foreground image $I^F_c = I \cdot M_c$ and category-background image $I^B_c = I \cdot (1 - M_c)$ by conducting element-wise multiplication “·”. The category-foreground and category-background image are both fed into the semantic encoder to produce the corresponding semantic embeddings. We apply different constraints to the two complementary images using the semantic embeddings and the category semantics to optimize the generation of $M_c$. For the category-foreground image, we apply category-foreground loss to the semantic encoder to ensure that the category-foreground image only contains the semantic of this category. For the category-background image, we apply category-background loss to the semantic encoder to ensure that the category-background image contains no information of this category. During training phase, the two complementary losses play an adversarial game, making the segmentation network aware of the semantic of this category and therefore can improve the correctness of its segmentation.

3.2.3 Loss Functions

In this section, we introduce the loss functions for training the segmentation network.

For the segmentation network, we design four losses (category-foreground, category-background, activation regularization, and consistency loss) to make the segmentation network aware of the category-specific semantic. As mentioned above, we construct an complementary image pair: category-foreground and category-background images for category-foreground loss and category-background loss, respectively. We provide some examples in Fig 4 for illustration. As shown in Fig 4, for a given image containing motorbike and person categories, the category-foreground map (i.e., the activation map) $M_c$ of category $c (c \in \{motorbike, person\})$ is element-wisely multiplied with the given image to produce the category-foreground image $I^F_c$ of category $c$. The category-background image $I^B_c$ of category $c$ is derived from the element-wise multiplication of the category-background map $1 - M_c$ and the given image. In the following sections, we introduce the four losses in details.

Category-Foreground Loss. The category-foreground (CF) loss aims at telling the segmentation network whether the activated regions in the category-foreground images are correct. As shown in Fig 4, for category $c$, the corresponding category-foreground image only contains regions of category $c$, the remaining regions of other categories are not activated. Let $E^F_c$ be the CSS of the category-foreground image for category $c$. Accordingly, the category-foreground loss is supposed to maximize the similarity between $E^F_c$ and $E_c$ and minimize the similarity between $E^F_{c'}$ and $E_{c'}$ where $c' \neq c$. We found that during the training of segmentation network, losses without log function produce better results.
Therefore, the CF loss $\mathcal{L}_{CF}$ is defined as:

$$
\mathcal{L}_{CF} = \frac{1}{C} \sum_{c}^{K} \mathbb{1}(Y_c = 1)[1 - \text{Sim}(E^F_c, \mathcal{E}_c)] + \frac{1}{C-1} \sum_{c', c' \neq c}^{K} \mathbb{1}(Y_c = 1, Y_{c'} = 1) \text{Sim}(E^F_{c'}, \mathcal{E}_{c'}),
$$

(11)

where $C$ is the number of available categories of the image.

**Category-Background Loss.** Similar with the CAM-based methods, the category-foreground loss encourages to activate the most discriminative regions. In this case, part of the objects will be included in the background. In view of this, we propose a category-background (CB) loss to address the issue of incomplete activation. The category-background loss aims to minimize the similarity between the semantic embedding of the category-foreground image $E^F_c$ and the category semantic $\mathcal{E}_c$, to ensure that the category-background image contains as little information of this category as possible. The category-background loss also encourages to activate the regions of other available categories of the image, by maximizing the similarity between the CSS of other categories $E^B_{c'}$ and the corresponding category semantics $\mathcal{E}_{c'}$, where $c' \neq c$. In this way, the category-background loss ensures that the the category-background image is complementary to the category-foreground image and only contains information of other categories available in the image. The CB loss $\mathcal{L}_{CB}$ is defined as:

$$
\mathcal{L}_{CB} = \frac{1}{C} \sum_{c}^{K} \mathbb{1}(Y_c = 1)[\text{Sim}(E^F_c, \mathcal{E}_c)] + \frac{1}{C-1} \sum_{c', c' \neq c}^{K} \mathbb{1}(Y_c = 1, Y_{c'} = 1)[1 - \text{Sim}(E^F_{c'}, \mathcal{E}_{c'})],
$$

(12)

where $C$ is the number of available categories of the image.

During training, the category-background loss plays an adversarial game with the category-foreground loss, which is similar to the erasing-based methods. However, the erasing-based methods often erase the image with binary mask using hard threshold. Besides, they usually randomly select a category of an image for training. The hard threshold will bring difficulty in optimization and the threshold need to be finely tuned. As Zhang et al. [37] indicated that, the training phase of the erasing-based approaches is unstable due to the randomness. In contrast, in our method, we directly multiply soft activation, instead of binary mask, with the image, and all available categories are taken into training phase. Therefore, we avoid the difficulty in optimization caused by the hard threshold, and training instability caused by the randomness.

**Activation Regularization Loss.** Through adversarial training, the category-foreground loss and category-background loss generate correct and complete activation maps. As the activated regions extend in the category-foreground image, the category-background loss decreases, until the entire object is included. If the activated regions become larger than the object, the category-foreground loss will increase. However, when backgrounds look similar with objects, e.g., a white boat surrounded by the white spindrift in the sea, the adversarial training of the two losses might converge to a trivial case, where large background regions are activated. In such a case, the activated regions are much larger than the object. To address this issue, we propose an activation regularization (AR) loss to compact the activated regions in the category-foreground image. The AR loss $\mathcal{L}_{AR}$ is defined as:

$$
\mathcal{L}_{AR} = \frac{\sigma}{KHW} \sum_{c}^{K} \sum_{i}^{H} \sum_{j}^{W} M_c(i, j),
$$

(13)

where $\sigma$ denotes the scaling value of the AR loss and set to 0.075 in our experiments.

**Consistency Loss.** In our observation, we find that some pixels of the category-foreground maps are activated in multiple channels. For example, for some train images, some pixels of the railway close to the train are activated in both background and train categories. In contrast, for some pixels of a black hole in a motorbike image are not activated in any channels. In these cases, the inconsistency occurs. To address this issue, we propose a consistency (CST) loss to ensure that each pixel is activated in only one of the channels. We first generate an background map for each category by aggregating the category-foreground maps of other channels. The background map $N_c$ for category $c$ indicates whether a pixel does not belong to this category, and its value for pixel $(i, j)$ is computed as:

$$
N_c(i, j) = max\{\{M_c(i, j) | c' \in \{1, \ldots, K\}, c' \neq c\}\},
$$

(14)

The consistency loss $\mathcal{L}_{CST}$ forces the summation of $M_c$ and $N_c$ to 1:

$$
\mathcal{L}_{CST} = \frac{1}{CHW} \sum_{c, Y_c = 1}^{K} \sum_{i}^{H} \sum_{j}^{W} (M_c(i, j) + N_c(i, j) - 1)^2,
$$

(15)

where $H$ and $W$ denote the height and width of the activation map. With the category-foreground loss and category-background loss, the segmentation network will choose the correct channel to activate for each pixel.

During training, the activation regularization loss plays another adversarial game with the category-background loss, since the activation regularization loss decreases the
activated regions in the category-foreground image and the category-background loss enlarges it. The adversarial training between the four losses ensures the correctness, completeness, compactness and consistency of the segmented regions, and thus boosts the segmentation performance.

Overall Loss Function  For training of the segmentation network, the overall loss function $L_{seg}$ is defined as:

$$L_{seg} = L_{CF} + L_{CB} + L_{AR} + L_{CST}.$$  \hfill (16)

4. Experiments

4.1. Dataset and Evaluation Metric

We conduct our experiments on PASCAL VOC 2012 dataset [6] with 21 categories (20 object categories and background category). As most approaches do, we use the augmented trainaug split (10528 images) with only image-level labels for training. The train split (1464 images) is used to validate our method. The val (1464 images) and test (1456 images) split are used to evaluate our WSSS approach and compare with other methods, respectively. We report all the experimental results in the standard mean Intersection over Union (mIoU) metric for semantic segmentation.

4.2. Implementation Details

For semantic encoder, we adopt the widely used VGG16 [24] pretrained on ImageNet [5] as the backbone. We use the stochastic gradient descent (SGD) optimizer to train the semantic encoder for 200 epochs with batch size 64. We adopt the poly-policy training strategy with the initial learning rate of 0.0025 and decay power of 0.9. The input images are randomly scaled and cropped to $224 \times 224$ pixels.

For segmentation network, we follow [13, 26, 28, 37] to use ResNet38 [31] pretrained on ImageNet [5] as the backbone for fair comparisons. We use the SGD optimizer to train the segmentation network for 8 epochs with batch size 8. We adopt the poly-policy training strategy with the initial learning rate of 0.01 and decay power of 0.9. The input images are randomly scaled and cropped to $448 \times 448$ pixels.

4.3. Ablation Studies

Table 1. The impact of each loss function of semantic encoder. Here $L_{GS}$, $L_{LS}$ and $L_{SC}$ represent the global similarity, local similarity and semantic correlation loss, respectively.

| Loss Function | $L_{GS}$ | $L_{LS}$ | $L_{SC}$ | mIoU (%) |
|---------------|----------|----------|----------|----------|
| SLAM (ours)   | ✓        | ✓        | ✓        | **57.4** (+3.0) |
| SLAM (ours)   | ✓        | ✓        | ✓        | **55.8** (+1.4) |
| SLAM (ours)   | ✓        | ✓        |           | 54.4 (+0.0) |

**Figure 5.** Visualization results for different loss functions of semantic encoder. The 2nd to 4th columns show the activation maps produced by different loss combinations.

4.3.1 The loss function of semantic encoder.

In this section, we evaluate the contribution of each loss function of the semantic encoder to the final segmentation performance. We show some visualization examples generated from different loss combinations in Fig. 5. It can be observed from the figure that when only global similarity loss is used, many object regions are not included in the activation map. With the addition of local similarity and semantic correlation loss, the activated regions becomes more and more complete. We also provide a quantitative result of different loss combinations in Table 1. It can be observed that the combination of all three losses yields the best mIoU of 57.4%.

**Figure 6.** Visualization results for different loss functions of segmentation network. The 2nd to 5th columns show the activation maps produced by different loss combinations.

4.3.2 The loss function of segmentation network.

In this section, we evaluate the impact of each loss function of the segmentation network to the final segmentation performance. We first show some visualization examples generated from different loss combinations in Fig 6. It can be observed that there exists two problems when only cat-
category foreground loss is used (the 2nd column of Fig. 6). First, the activated regions are incomplete, e.g., only a part of the boat is activated. Second, the background regions such as the spindrift in the sea are falsely activated. For the first problem, the addition of category-background loss $L_{CB}$ successfully improve the completeness of the activated regions, by removing the background information in the category-foreground image. The segmentation network trained with $L_{CF} + L_{CB}$ obtains more complete regions than that trained with category-foreground loss $L_{CF}$. However, as mentioned above, there exists a trivial case where the adversarial training between $L_{CF}$ and $L_{CB}$ fails, i.e. activating a very large region in the category-foreground image (e.g. the 3rd column of Fig 6). The AR addresses this issue by shrinking the area of the activated regions, therefore, the segmented regions of the objects become more compact. The consistency loss can alleviate the second problem. With the help of the other losses, the consistency loss forces the segmentation network to activate only one correct channel for each pixel. As shown in Fig 6, the combination of the four losses can decrease the false activation e.g. the railway close to the train in the 4th row. Table 2 lists the segmentation performance of different combinations of the loss functions. As shown in Table 2, the combination of the four losses achieves the best mIoU of 57.4%.

### 4.4. Main Results

#### 4.4.1 Performance of CAM and pseudo mask

We first compare the segmentation performance of the CAM and pseudo mask in Table 3. As shown in the 2nd column of the table, our SLAM achieves 57.4% mIoU, which is 10.0% higher than that of CAM [39], and surpasses many recent binary image-level classification based approaches, e.g. 1.8% and 0.8% higher than that of AdvCAM [15] and ECS-Net [26], respectively. We also compare the performance of the pseudo mask generated by AffinityNet [1], which is widely adopted by recent methods. As shown in the last column of the table, our pseudo mask achieves the best mIoU of 69.9% among the listed approaches.

We now compare the performance of WSSS between our method and recent approaches in Table 4. For the fully supervised segmentation network, we follow OC-CSE [13] and CPN [37] to use DeeplabV1 [3] with ResNet38 [31] backbone. As shown in the table, our SLAM achieves 71.1% and 71.5% mIoU on PASCAL VOC 2012 val and test split, respectively, outperforming many recent binary image-level classification based approaches by around 3% mIoU, such as OC-CSE [13] and ECS-Net [26]. Fig 7 shows some examples of our WSSS on val split. Our SLAM also surpasses recent proposed CLIMS [33] utilizing extra...
multi-modal knowledge, and AMN [16] trained with strong per-pixel constraint.

Figure 7. Visualization results of WSSS on PASCAL VOC 2012 val split. From top to bottom: Image, our results, and Ground Truth.

5. Conclusions

In this paper, we propose a novel framework Semantic Learning based Activation Map (SLAM) for weakly supervised semantic segmentation. We design a semantic encoder to learn the semantic of each category. We use the semantic embeddings of the foreground and background to supervise the generation of activation maps under the constraint of four losses (category-foreground, category-background, activation regularization, and consistency loss). Experiments show that our SLAM achieves much better performance than binary image-level classification approaches, and surpasses recent approaches trained with strong per-pixel constraint or multi-modal knowledge.

References

[1] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In CVPR, pages 4981–4990, 2018. 3, 8
[2] Amy L. Bearman, Olga Russakovsky, Vittorio Ferrari, and Fei-Fei Li. What’s the point: Semantic segmentation with point supervision. In ECCV, pages 549–565, 2016. 1
[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. ICLR, 2015. 1, 8
[4] Chen, Liang-Chieh, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. TPAMI, 40(4):834–848, 2018. 1
[5] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In CVPR, pages 248–255, 2009. 7
[6] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. IJCV, 111(1):98–136, 2015. 7
[7] Junsong Fan, Zhaoxiang Zhang, and Tieniu Tan. Employing multi-estimations for weakly-supervised semantic segmentation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, ECCV, pages 332–348, 2020. 8
[8] Junsong Fan, Zhaoxiang Zhang, Tieniu Tan, Chunfeng Song, and Jun Xiao. Cian: Cross-image affinity net for weakly supervised semantic segmentation. In AAAI, pages 10762–10769, 2020. 3
[9] Zilong Huang, Xinggang Wang, Jiashi Wang, Wenyu Liu, and Jingdong Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In CVPR, pages 7014–7023, 2018. 3
[10] Peng-Tao Jiang, Qibin Hou, Yang Cao, Ming-Ming Cheng, Yunchao Wei, and Hong-Kai Xiong. Integral object mining via online attention accumulation. In ICCV, pages 2070–2079, 2019. 8
[11] Anna Khoreva, Rodrigo Benenson, Jan Hendrik Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In CVPR, pages 1665–1674, 2017. 1
[12] Alexander Kolesnikov and Christoph H. Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In ECCV, pages 695–711, 2016. 3
[13] Hyeokjun Kweon, Sung-Hoon Yoon, Hyenonseong Kim, Daeehee Park, and Kuk-Jin Yoon. Unlocking the potential of ordinary classifier: Class-specific adversarial erasing framework for weakly supervised semantic segmentation. In ICCV, pages 6994–7003, 2021. 1, 2, 3, 7, 8
[14] Jungbeom Lee, Jooyoung Choi, Jisoo Mok, and Sunghro Yoon. Reducing information bottleneck for weakly supervised semantic segmentation. In Marc’Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, NeurIPS, pages 27408–27421, 2022. 1, 2, 3
[15] Jungbeom Lee, Eunji Kim, and Sunghro Yoon. Anti-adversarially manipulated attributions for weakly and semi-supervised semantic segmentation. In CVPR, pages 4071–4080, 2021. 8
[16] Minhyun Lee, Dongseob Kim, and Hyunjung Shim. Threshold matters in WSSS: manipulating the activation for the robust and accurate segmentation model against thresholds. In CVPR, 2022. 1, 2, 3, 8, 9
[17] Kunpeng Li, Ziyu Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Tell me where to look: Guided attention inference network. In CVPR, 2018. 1, 2, 3
[18] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In CVPR, pages 3159–3167, 2016. 1
[19] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, ECCV, pages 740–755, 2014. 8
[20] Yun Liu, Yu-Huan Wu, Pei-Song Wen, Yu-Jun Shi, Yu Qiu, and Ming-Ming Cheng. Leveraging instance-, image- and dataset-level information for weakly supervised instance segmentation. TPAMI, 2020. 8
[21] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3431–3440, 2015. 1

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In ICML, pages 8748–8763, 2021. 3

[23] Wataru Shimoda and Keiji Yanai. Self-supervised difference detection for weakly-supervised semantic segmentation. In ICCV, pages 5207–5216, 2019. 3

[24] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015. 7

[25] Chunfeng Song, Yan Huang, Wanli Ouyang, and Liang Wang. Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation. In CVPR, pages 3136–3145, 2019. 1

[26] Kunyang Sun, Haoqing Shi, Zhengming Zhang, and Yongming Huang. Ecs-net: Improving weakly supervised semantic segmentation by using connections between class activation maps. In ICCV, pages 7283–7292, 2021. 1, 7, 8

[27] Paul Vernaza and Manmohan Chandraker. Learning random-walk label propagation for weakly-supervised semantic segmentation. In CVPR, 2017. 1

[28] Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In CVPR, pages 12275–12284, 2020. 2, 3, 7, 8

[29] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In CVPR, pages 1568–1576, 2017. 1, 2

[30] Yunchao Wei, Huaxin Xiao, Honghui Shi, Zequn Jie, Jiashi Feng, and Thomas S. Huang. Revisiting dilated convolution: A simple approach for weakly- and semi-supervised semantic segmentation. In CVPR, pages 7268–7277, 2018. 2

[31] Zifeng Wu, Chunhua Shen, and Anton Van Den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. PR, pages 119–133, 2019. 7, 8

[32] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In Proceedings of the European conference on computer vision (ECCV), pages 325–341, 2018. 1

[33] Dong Zhang, Hanwang Zhang, Jinhui Tang, Xian-Sheng Hua, and Qianru Sun. Causal intervention for weakly-supervised semantic segmentation. In NeurIPS, 2020. 8

[34] Fei Zhang, Chaochen Gu, Chenyue Zhang, and Yuchao Dai. Complementary patch for weakly supervised semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 7242–7251, 2021. 1, 2, 3, 6, 7, 8

[35] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yahiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In CVPR, pages 6881–6890, 2021. 1

[36] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In CVPR, pages 2921–2929, 2016. 2, 8