Noname manuscript No.
(will be inserted by the editor)

Weakly-Supervised Semantic Segmentation with Visual Words Learning and Hybrid Pooling

Lixiang Ru · Bo Du · Yibing Zhan · Chen Wu

Received: date / Accepted: date

Abstract Weakly-Supervised Semantic Segmentation (WSSS) methods with image-level labels generally train a classification network to generate the Class Activation Maps (CAMs) as the initial coarse segmentation labels. However, current WSSS methods still perform far from satisfactorily because their adopted CAMs 1) typically focus on partial discriminative object regions and 2) usually contain useless background regions. These two problems are attributed to the sole image-level supervision and aggregation of global information when training the classification networks. In this work, we propose the visual words learning module and hybrid pooling approach, and incorporate them in classification network to mitigate the above problems. In visual words learning module, we counter the first problem by enforcing the classification network to learn fine-grained visual word labels so that more object extents could be discovered. Specifically, the visual words are learned with a codebook, which could be updated via two proposed strategies, *i.e.* learning-based strategy and memory-bank strategy. The second drawback of CAMs is alleviated with the proposed hybrid pooling, which incorporates the global average and local discriminative information to simultaneously ensure object completeness and reduce background regions. We evaluated our methods on PASCAL VOC 2012 and MS COCO 2014 datasets. Without any extra saliency prior, our method achieved 70.6% and 70.7% mIoU on the *val* and *test* set of PASCAL VOC dataset, respectively, and 36.2% mIoU on the *val* set of MS COCO dataset, which significantly surpassed the performance of state-of-the-art WSSS methods.

Keywords Weakly-Supervised Semantic Segmentation · Visual Words Learning · Hybrid Pooling · Semantic Segmentation

Lixiang Ru [1]
E-mail: rulixiang@whu.edu.cn

✉ Bo Du [1] (*Corresponding Author.*)
E-mail: dubo@whu.edu.cn

Yibing Zhan [2]
E-mail: zhanyibing@jd.com

Chen Wu [3]
E-mail: chen.wu@whu.edu.cn

[1] School of Computer Science, Wuhan University, Wuhan, China
[2] JD Explore Academy, JD.com, Beijing, China
[3] LIESMARS, Wuhan University, Wuhan, China

1 Introduction

Semantic segmentation, aiming at assigning a specific label for each pixel in an image, is a fundamental and hot topic in computer vision. With the rapid development of deep learning, semantic segmentation based on deep neural networks has dominated the past decades (Long et al., 2015; Chen et al., 2015; Badrinarayanan et al., 2017; Chen et al., 2017). However, the data-hungry nature of deep models determines that to obtain a segmentation model with fancy performance, a large number of images with well-annotated pixel-level labels are indispensable. Unfortunately, pixel-level labels are usually very costly in both time and money. The empirical statistics in (Lin et al., 2019) show that annotating the pixel-level label of an image in the PASCAL VOC dataset (Everingham et al., 2010) needs about 4 minutes on average, meanwhile annotating the Cityscapes dataset (Cordts et al., 2016) takes an even longer time, about 90 minutes per image.

To address the above problem, many researchers have dedicated to devising image segmentation models

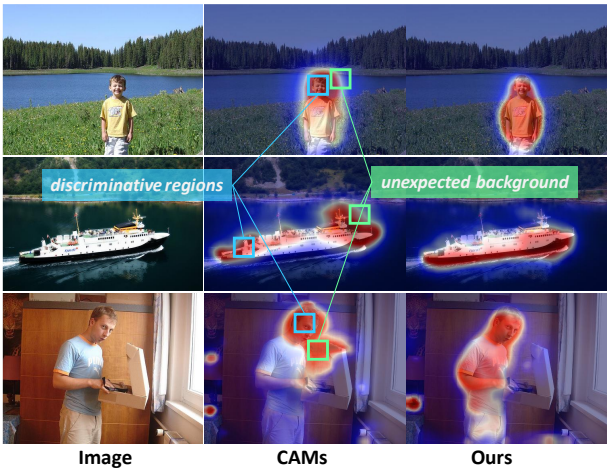


Fig. 1: Illustration of the drawbacks of CAMs. Typically, CAMs only discover partial discriminative object regions and adjacent background regions. We argue that these drawbacks are attributed to the sole image-level supervision and aggregation of global information. To mitigate them, in this work, we proposed the visual words learning and hybrid pooling module.

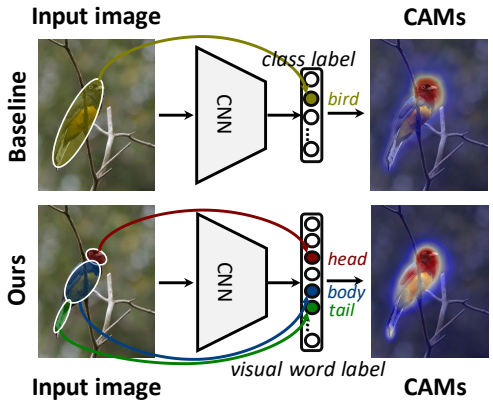


Fig. 2: Illustration of the motivation of our visual words learning module.

with weaker and cheaper labels, such as image-level labels (Papandreou et al., 2015; Pinheiro and Collobert, 2015; Ahn and Kwak, 2018; Lee et al., 2021a). Prevailing WSSS methods with image-level labels usually adopt a multi-step framework. Specifically, these WSSS methods firstly train classification networks with only image-level labels and use the trained classification networks to generate initial coarse pixel-level labels by class activation mapping (Zhou et al., 2016). Then, the coarse pixel-level labels will be further refined by methods like dense CRF (Krähenbühl and Koltun, 2011) and other pixel affinity-based approaches (Ahn and Kwak, 2018; Ahn et al., 2019) to obtain the refined pseudo labels. Finally, the refined pseudo labels are used to train a regular semantic segmentation model to predict pixel-level labels of test images.

Prior works have demonstrated that the first step, *i.e.* generating initial coarse labels, is crucial to the training of segmentation models and the final segmentation performance (Wang et al., 2020b; Chang et al., 2020b; Lee et al., 2021a). As aforementioned, most methods train classification networks to produce Class Activation Maps (CAMs) (Zhou et al., 2016) as the initial coarse labels. However, as illustrated in Fig. 1, there are two typical drawbacks of previous CAMs. Firstly, CAMs usually only discover partial discriminative regions of visual objects. The reason is that CAMs are derived from classification networks, whose purpose is to differentiate different semantic categories. Therefore, to attain discriminability, the classification network will shift attention to the most discriminative regions of visual objects instead of the integral object. Secondly, the activated regions of CAMs often include some undesired background. This is attributed to that classification networks commonly use global average pooling (GAP) (Lin et al., 2013) for feature aggregation, which averages information from both foreground objects and background, thus overestimating the size of objects (Zhou et al., 2016).

To tackle the first problem, as illustrated in Fig. 2, we argue that if the network could be supervised by more fine-grained labels, more object regions will be activated to provide sufficient information for differentiating different classes. Therefore, in this work, we propose the Visual Words Learning (VWL) module for WSSS task with image-level labels. The VWL module generates the visual word labels by using a codebook to encode the feature maps extracted by the CNN backbone. In the training process of the classification network, the network will be forced to jointly learn the image-level labels and visual word labels so that more object regions could be activated. To learn an effective codebook, based on the definition and solution of Bag of Visual Word models (BoVW) (Arandjelović et al., 2017; Passalis and Tefas, 2017), we devised two strategies for updating the codebook, *i.e.* learning-based strategy and memory-bank strategy. For the learning-based strategy, the codebook is set as a learnable parameter. By enforcing the encoded visual word features to learn image-level labels, the codebook could learn the latent visual word representations. In practice, we also notice that the learned representations in the codebook are often redundant, which affects the network training and the quality of CAMs. We tackle this problem by regularizing the codebook with DeCov loss (Cogswell et al., 2017), which reduces the redundancy of a matrix by minimizing its off-diagonal

co-variance values. For the memory-bank strategy, we follow the classic BoVW models (Liu et al., 2019; Gidaris et al., 2020), which take the clustering centroids of features as the codebook. Specifically, we decompose the clustering on the whole training set to each mini-batch iteration and leverage memory-bank strategy (Wu et al., 2018; Zhuang et al., 2019) to gradually update the codebook. Our experimental results show that, after sufficient updates, the learning-based and memory-bank strategy could both learn codebooks with effective representations of visual words and achieve analogous performance.

To alleviate the second drawback, inspired by global max-pooling (GMP), which takes the maximum value in each feature map as outputs and tends to underestimate the object sizes (Kolesnikov and Lampert, 2016), we proposed a simple yet empirically effective pooling approach, named Hybrid Pooling (HP). Our major motivation of HP is to aggregate the local maximum values so that less background information is involved. In specific, the feature maps are partitioned into multiple bins from coarse to fine levels. For bins in the same level, we pool them separately via max pooling and average the aggregated features so that only local maximums are involved. The features from different levels and the output feature of GAP (to ensure the object completeness) are then averaged as the final outputs. On this account, more discriminative object extents and fewer background regions are preserved in feature maps, which could improve the accuracy of the generated CAMs.

We conducted extensive experiments to verify the effectiveness of our proposed methods. The experimental results showed that our method significantly improved the quality of generated CAMs. We refined the generated CAMs with IRNet (Ahn et al., 2019) and trained a DeepLabV2 segmentation network (Chen et al., 2017) with the refined pseudo labels. Semantic segmentation results on two datasets, PASCAL VOC 2012 (Everingham et al., 2010) and MS COCO 2014 (Lin et al., 2014), showed that our proposed method could outperform the state-of-the-art methods.

Overall, our contributions in this work are summarized as follows.

- We propose the Visual Words Learning (VWL) module. By jointly learning the visual word labels and image-level labels, the network is enforced to discover integral object extents. To encode the visual words, we devise two learning strategies to learn the codebook and empirically verify their efficacy.
- We propose Hybrid Pooling (HP), a simple yet effective pooling approach, which incorporates local discriminative information and global information to aggregated less background and more object regions.
- By incorporating the proposed VWL and HP, we present a new classification network to generate CAMs with higher quality for the WSSS task. Our method achieves new state-of-the-art performance, *i.e.* 70.6% mIoU on PASCAL VOC 2012 *val* set and 36.2% mIoU on MS COCO 2014 *val* set.

This paper is an improved version of our preliminary work (Ru et al., 2021). Compared with the conference version, this work further improves the learning-based strategy and proposes the memory-bank strategy which could learn visual words better. The performance is remarkably improved with these improvements and surpasses the latest state-of-the-art methods. We also conduct further experiments on more datasets to verify the efficacy of our approach.

The rest of this paper is structured as follows. In Section 2, we briefly introduce some related works on WSSS with image-level labels and improvements on CAMs. The detailed methods are presented in Section 3. We present the experimental settings and results in Section 4. Section 5 concludes our work.

2 Related Work

2.1 WSSS with Image-level Labels

Weakly-Supervised Semantic Segmentation (WSSS) aims to develop semantic segmentation models with weak annotations, such as image-level labels (Papandreou et al., 2015; Pinheiro and Collobert, 2015; Ahn and Kwak, 2018; Lee et al., 2021a), points (Bearman et al., 2016), bounding boxes(Song et al., 2019; Oh et al., 2021; Lee et al., 2021b), and scribbles (Lin et al., 2016). In this work, we focus on WSSS with only image-level labels. In the subsections below, we will introduce WSSS methods with image-level labels based on their motivations.

Growing Seed Regions with Constraints. (Kolesnikov and Lampert, 2016) proposed *SEC* principle to expand the initial seed cues and coincide with the object shapes. This principle was adopted by subsequent works. For example, (Roy and Todorovic, 2017) used CRF-CNN (Zheng et al., 2015) to refine the initial labels with low-level pixel information to generate pseudo labels fitting object boundaries. (Huang et al., 2018) integrated seeded region growing (Adams and Bischof, 1994) to expand the initial seed cues generated from classification networks and also adopted dense CRF (Krähenbühl and Koltun, 2011) to refine pseudo labels.

Erasing. Based on the common observation that CAMs usually only captured the most discriminative regions, Wei *et al*. proposed to adversarially erase the discriminative regions and progressively localize the integral object regions (Wei et al., 2017). Similarly, ACoL (Zhang et al., 2018) used two parallel CNN to erase the feature maps in one branch with the discriminative regions derived from the other branch and fused the localized regions from both branches as outputs. To prevent the attention regions from shifting to non-object regions during erasing, SeeNet (Hou et al., 2018) used saliency prior (Hou et al., 2017) to suppress the attention in background regions.

Accumulating Attention. Another interesting observation is that classification networks tend to shift attention to the different regions of the object across the training process (Jiang et al., 2019). Motivated by this, Jiang *et al*. proposed OAA, which accumulated the activated regions during the different training stage. In (Yao et al., 2021), Yao *et al*. proposed a graph reasoning and non-salient region mining module to capture more object extents from non-salient regions, since the saliency prior used in OAA did not always correspond to the foreground objects. Kim *et al*. in (Kim et al., 2021) combined the idea of erasing and accumulating to suppress the discriminative regions in training, which could assist in finding less discriminative object regions.

Mining Objects from Multiple Images. The works above mainly focused on mining semantic objects from single image. Some recent works also tried to leverage the semantic co-occurrence in two or more images (Fan et al., 2020; **?**; Sun et al., 2020). In (Fan et al., 2020), CIAN designed a cross image attention module to model the pixel-level affinity from different images with common semantics. In (Sun et al., 2020), in addition to the co-attention from image pairs, Sun *et al*. further proposed a contrastive attention module that could mine the unique semantic objects. (**?**) used a graph neural network (GNN) (Scarselli et al., 2008) based approach to reason and capture the integral object information from a group of input images.

Refining Seed Regions. Since CAMs only yield very coarse pixel labels, some pixel affinity based methods are proposed to refine CAMs and proved to work brilliantly. (Wang et al., 2020a) proposed an EM framework, in which a unary network was used to predict the class score maps and a pairwise network was used to learn the pixel affinities. The learned pixel affinity would be used to refine the score maps and then supervise the training of the framework. PSA (Ahn and Kwak, 2018) derived foreground and background regions with high confidence from coarse labels and utilized the reliable regions to learn an affinity network. For each input image, their coarse labels were refined via random walk propagation (Vernaza and Chandraker, 2017) with the learned affinity matrix from the trained network. In further, IRNet (Ahn et al., 2019) proposed to derive instance labels from instance-agnostic CAMs via additionally learning semantic instance boundaries and propagating the initial CAMs with the learned pixel affinities and instance boundaries.

End-to-End Solutions. Though the majority of methods adopted a multi-step framework, some works also tried to devise elegant end-to-end models for WSSS with image-level labels. In (Papandreou et al., 2015), Papandreou *et al*. proposed an EM framework that estimated the pseudo labels with a probabilistic model and utilized the pseudo labels to train the network at the maximization step. (Zhang et al., 2020a) followed a similar framework of (Papandreou et al., 2015) but leveraged CRF to refine CAMs and jointly minimized the cross-entropy loss and low-level energy loss with the highly-confident pseudo labels. (Araslanov and Roth, 2020) devised normalized global weighted pooling to aggregate classification scores from predicted score maps, which could improve the completeness of objects. The predicted score maps were then refined with a pixel affinity-based module to supervise the training process. To avoid the network degrading to trivial solutions, (Araslanov and Roth, 2020) additionally proposed a stochastic gate that randomly transferred low-level features to high-level semantics.

2.2 Improvements on CAMs

A prevailing series of WSSS with image-level labels is to derive better CAMs from the classification network. Many works have been proposed to produce better CAMs by encouraging more object extents to be discovered. In (Wang et al., 2020b), Wang *et al*. observed that CAMs of the same image with different scaling ratio usually differed largely in shape, while they were supposed to be the same since they consisted of the same objects. They proposed to regularize the classification network by minimizing the difference between the CAMs of different scales and achieved remarkable performance. In (Chang et al., 2020b), Chang *et al*. proposed to cluster the original semantic categories to sub-categories and further leveraged the sub-category labels to supervise the training of the network, which could enforce the network to discover more object regions to distinguish sub-categories. (Lee et al., 2021a)

proposed an anti-adversarial attack approach to gradually pull image away from decision boundaries which helped to discover more object regions. (Zhang et al., 2020b) firstly introduced causal inference (Rubin, 2019) to alleviate the confounding bias attributed by ambiguous boundaries. Some recent works showed that data augmentation techniques (Chang et al., 2020a) and auxiliary self-supervised tasks (Jo and Yu, 2021) could help to discover more object regions and thus improve the quality of CAMs. In this work, we also focus on deriving better CAMs from classification networks but from two aspects. Specifically, we propose the visual words learning module and hybrid pooling approach to counter the problem of partial discriminative object regions unexpected background regions in CAMs.

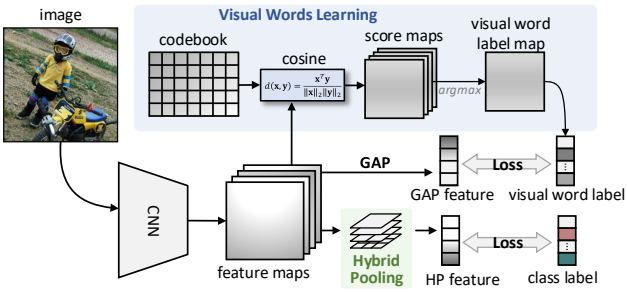


Fig. 3: Overview of the proposed method. To encourage more object extents to be discovered, we propose the Visual Words Learning (VWL) module, which utilizes a codebook to encode the feature maps extracted by CNN. The encoded visual word labels are then used to supervise the training process of the classification network. We also propose a novel feature aggregation method, *i.e.* Hybrid Pooling (HP), which incorporates GMP to reduce background information and GAP to ensure the object completeness in the generated CAMs.

3 Methods

This section expatiates our proposed methods, including the Visual Words Learning (VWL) module, Hybrid Pooling (HP) approach, and training process of our network.

3.1 Method Overview

The overall architecture of our method is presented in Fig. 3. For an input image, we firstly use a CNN backbone to extract the convolutional feature maps. In the visual words learning module, a predefined codebook is employed to encode the feature maps to visual word score maps, in which each element denotes the probability of a pixel belonging to each visual word. The visual word label of a given image is derived from the score maps and further used to supervise the network training to activate more object regions. To alleviate the problem of unexpected background regions, we use the proposed hybrid pooling, which aggregates local discriminative information and global average information, so that less background is preserved in the generated CAMs.

3.2 Preliminaries

Currently, the majority of WSSS methods with image-level labels infer CAMs (Zhou et al., 2016) from a trained classification network as the initial coarse labels. In this work, we follow the original way in (Zhou et al., 2016) to directly produce CAMs with the feature maps of the last *conv* layer and the weight matrix $\mathbf{W}^{img}$ in prediction layer. Specifically, CAMs for class c are given by weighting each feature map in $\mathbf{F}$ with its contribution to class c

$$\mathbf{M}_c = \sum_{i=1}^{d} (\mathbf{W}_{i,c}^{img} \mathbf{F}_{:,:,i}). \tag{1}$$

$\mathbf{M}_c$ is further passed through a *relu* layer to eliminate the negative values, denoted as $\hat{\mathbf{M}}_c$. The generated $\hat{\mathbf{M}}_c$ will be used to produce pseudo segmentation labels with a background score threshold.

3.3 Visual Words Learning

As aforementioned, classification networks guided by image-level labels usually only discover partial discriminative extents of objects. The reason is that focusing on partial discriminative extents of objects is more beneficial to recognize different semantic categories. To solve this problem, our motivation is that if the network could be supervised with more fine-grained labels in the training procedure, it will be enforced to activate more semantic regions so that the generated CAMs would be more accurate. To this end, we propose to jointly learn the visual words and image-level labels in the training process of classification networks.

Since only image-level labels are available in our task, in order to leverage visual word labels to guide the training of classification networks, we design an unsupervised visual words learning module. As shown in Fig. 3, in the visual words learning module, a matrix $\mathbf{C} \in \mathbb{R}^{k \times d}$ is defined as the codebook, where k is the number of words and d denotes the feature dimension. $\mathbf{C}$ is utilized to encode the extracted convolutional feature map $\mathbf{F} \in \mathbb{R}^{h \times w \times d}$ to specific visual words. Here,

we use cos distance to measure the similarity between the pixel at position i in $\mathbf{F}$ and the j-th word in $\mathbf{C}$. The similarity matrix $\mathbf{S}$ is given as:

$$\mathbf{S}_{ij} = \frac{\mathbf{F}_i^\top \mathbf{C}_j}{||\mathbf{F}_i||_2||\mathbf{C}_j||_2}, 1 \leq i \leq hw, 1 \leq j \leq k. \tag{2}$$

The obtained $\mathbf{S}$ will be normalized row-wisely using $softmax$ to compute the probability of the i-th pixel in $\mathbf{F}$ belonging to j-th word in codebook $\mathbf{C}$. The process is given as

$$\mathbf{P}_{ij} = \frac{\exp(\tau \cdot \mathbf{S}_{ij})}{\sum_{n=1}^{k} \exp(\tau \cdot \mathbf{S}_{in})}, \tag{3}$$

where τ>0 is a temperature parameter to control the smoothness of $\mathbf{P}$.

The visual word label $\mathbf{Y}_i$ for $\mathbf{F}_i$ is then given as the word with the maximum probability, *i.e.*, the index of the maximum value in the i-th row of $\mathbf{P}_{ij}$, which is denoted as

$$\mathbf{Y}_i = \arg\max_j \mathbf{P}_{ij}\,. \tag{4}$$

For an input image $\mathbf{X}$, its visual word labels are computed as a k-dimensional vector $\mathbf{y}^{word}$, where $\mathbf{y}_j^{word} = 1$ if the j-th word exists in $\mathbf{Y}$, and $\mathbf{y}_j^{word} = 0$, otherwise. $\mathbf{y}^{word}$ will be used to guide the training procedure of the classification network to enforce it to discover more discriminative extents.

Another problem is to ascertain the codebook for encoding visual word labels reasonably. In a classic BoVW model, the codebook is usually identified as the clustering centroids of the feature representations extracted from all local visual words (Liu et al., 2019; Gidaris et al., 2020). However, in our model, the feature representations for visual words are updated online as the training procedure. Therefore, the codebook $\mathbf{C}$ should also be updated online. To this end, as shown in Fig. 4, we devised two strategies to learn $\mathbf{C}$, namely the *Learning-based strategy* and *Memory-bank strategy*.

Learning-based strategy. In the learning-based strategy, following (Passalis and Tefas, 2017; Arandjelović et al., 2017), we set the codebook as a trainable parameter to learn it from back-propagated gradients. In a BoVW model, the frequencies of visual words are collected as the feature descriptor to predict the image classes so as to learn the relations between visual words and semantic classes (Liu et al., 2019). However, this hard quantization approach will introduce non-continuities and is proved to make the training process intractable (Passalis and Tefas, 2017). In this work, we compute the frequency of each word by accumulating the probabilities in $\mathbf{P}$. Therefore, the soft frequency assignment of the j-th word is

$$\mathbf{f}_j^{word} = \frac{1}{hw}\sum_{i=1}^{hw} \mathbf{P}_{ij}, \tag{5}$$

where $\mathbf{f}_j^{word}$ denotes the frequency of the j-th word in $\mathbf{F}$. As shown in Fig. 4 (a), $\mathbf{f}^{word}$ will be used to predict the image-level labels, *i.e.*, modeling the relations between visual words and image-level labels, which encourages the codebook to learn latent visual word representations via gradients.

Memory-bank strategy. As aforementioned, a classic BoVW model usually takes the clustering centroids of image features as the codebook. However, clustering on the whole dataset with the network training is extremely time-consuming. Inspired by mini-batch K-Means (Sculley, 2010), which decomposes clustering on a large-scale dataset to mini-batch iterations, we propose the memory-bank strategy to gradually update the codebook with reconstructed codebook in each training step.

In Eq. (4), the visual word label $\mathbf{Y}$ for each pixel in $\mathbf{F}$ is computed. We firstly transform $\mathbf{Y} \in \mathbb{R}^{hw}$ to $\mathbf{W} \in \mathbb{R}^{hw \times k}$ via one-hot encoding. The reconstructed codebook $\mathbf{C}'$ is hereby given as averaging the representations that are encoded to the same visual word category.

$$\mathbf{C}' = \mathbf{D}^{-1}\mathbf{W}^\top \mathbf{F}, \tag{6}$$

where $\mathbf{D}$ is a degree matrix with $\mathbf{D}_{ii} = 1/\sum_{j,k=i} \mathbf{W}_{jk}$ and $\mathbf{D}_{ij} = 0$ for off-diagonal elements. Here, we also assume $\mathbf{F}$ is unfolded to $\mathbb{R}^{hw \times d}$ for simplicity. As shown in Fig. 4 (b), the reconstructed codebook $\mathbf{C}'$ is then used to update codebook with a momentum parameter ρ. This process is given as

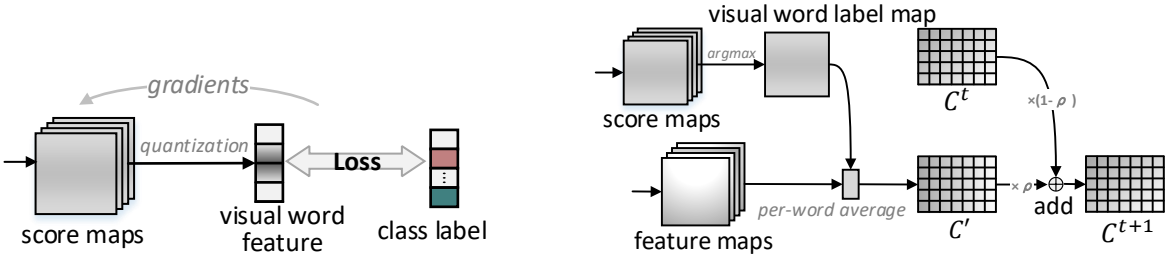
$$\mathbf{C}^{t+1} \leftarrow \rho\mathbf{C}' + (1-\rho)\mathbf{C}^t, \tag{7}$$

where t denotes the training iterations.

3.4 Hybrid Pooling

To mitigate the aforementioned disadvantages of GAP and GMP, in this work, we present a simple yet empirically effective pooling method which aggregates local maximum and global average values of the feature maps.

Considering the output feature map $\mathbf{F}$ with size of $h \times w \times d$ of the last convolutional layer, we firstly partition $\mathbf{F}$ to multi-scale divisions. As illustrated in Fig. 5,



(a) Learning-based strategy (b) Memory-bank strategy

Fig. 4: Illustration of strategies for visual word codebook. For the learning-based strategy, we follow the original intention of BoVW models (Liu et al., 2019; Gidaris et al., 2020), *i.e.* using visual word frequencies to predict the image-level labels, which could enforce to learn the codebook from the back-propagated gradients. For the memory-bank strategy, inspired by mini-batch K-means (Sculley, 2010), we decompose the clustering on the whole dataset to each training step and update the codebook from reconstruction in the memory-bank mechanism.

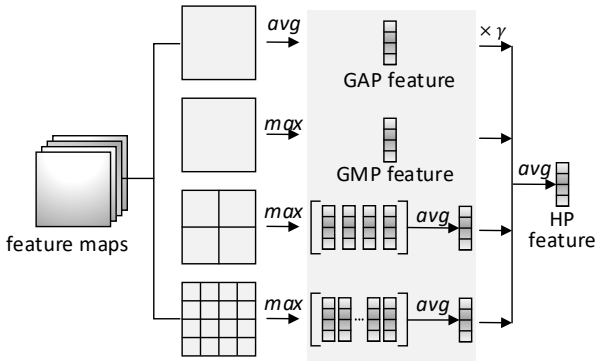


Fig. 5: Illustration of the proposed hybrid pooling. *avg*: average pooling, *max*: max pooling.

each division with size of $\frac{h}{r} \times \frac{w}{r} \times d$ is pooled to a d-dimensional vector via **max** pooling, where $r \in \{1, 2, 4\}$ denotes the split size. $\mathbf{F}$ is thus aggregated to $\mathbf{F}^{max}$ with size of $r \times r \times d$. It is conspicuous that $\mathbf{F}^{max}$ only involves local maximum pixels so that less background is considered. We then pool $\mathbf{F}^{max}$ for subsequent classification task. The pooled feature $\mathbf{f}_r^{max}$ with split size r is given by

$$\mathbf{f}_r^{max} = \frac{1}{r^2} \sum_{i=1}^{r} \sum_{j=1}^{r} \mathbf{F}_{i,j,:}^{max}. \tag{8}$$

Note that $\mathbf{f}_r^{max}$ only preserves the maximum responses of local objects, which may corrupt the completeness of objects. To tackle this problem, in HP module, we also incorporate the results of GAP. Given the pooled feature of GAP layer, which is computed as

$$\mathbf{f}^{gap} = \frac{1}{hw} \sum_{i=1}^{h} \sum_{j=1}^{w} \mathbf{F}_{i,j,:}, \tag{9}$$

the final output of hybrid pooling module is calculated by weighting the outputs in Eq. (8) and Eq. (9), computed as

$$\mathbf{f}^{hp} = \frac{1}{\gamma + 3} \Big(\sum_{r \in \{1,2,4\}} \mathbf{f}_r^{max} + \gamma \mathbf{f}^{gap} \Big), \tag{10}$$

where γ is a weight factor. Leveraging Eq. (10), more regions of foreground objects and less background are captured for classification, so that the generated CAMs could coincide better with object shapes.

3.5 Network Training

Since only image-level labels are available, the classification loss is indispensable to train the network. After calculating $\mathbf{f}^{hp}$ via Eq. (10), the classification score for image label is computed with a classification layer (an 1×1 *conv* layer in practice), denoted as $\mathbf{p}^{img} = conv(\mathbf{f}^{hp}, \mathbf{W}^{img})$, where $\mathbf{W}^{img}$ is the weight matrix of this layer. As a common practice (Wang et al., 2020b; Chang et al., 2020b; Araslanov and Roth, 2020), the multi-label soft margin loss (Paszke et al., 2019) is employed to compute the classification loss

$$\begin{aligned} \mathcal{L}_{cls}(\mathbf{p}^{img}, \mathbf{y}^{img}) = \frac{1}{L} \sum_{i=1}^{L} [\mathbf{y}_i^{img} \log \frac{\exp(\mathbf{p}_i^{img})}{1 + \exp(\mathbf{p}_i^{img})} \\ + (1 - \mathbf{y}_i^{img}) \log \frac{1}{1 + \exp(\mathbf{p}_i^{img})}], \end{aligned} \tag{11}$$

where $\mathbf{y}^{img}$ denotes the ground-truth image label and L is the number of image classes.

To capture more semantic regions, the pooled feature is also used to predict the visual word label $\mathbf{y}^{word}$ generated in previous steps. It is noted that $\mathbf{y}^{word}$ is generated based on all pixels in feature map $\mathbf{F}$. Therefore, we use GAP here instead of our HP to perform feature aggregation for predicting y^{word}. The predicted visual word score is thus denoted as $\mathbf{p}^{word} = conv(\mathbf{f}^{gap}, \mathbf{W}^{word})$,

where $\mathbf{W}^{word}$ is the weight matrix of the prediction layer. The classification loss for visual words is then denoted as $\mathcal{L}_{cls}(\mathbf{p}^{word}, \mathbf{y}^{word})$, which is in the same form as Eq. (11). The overall loss function is the sum of $\mathcal{L}_{cls}(\mathbf{p}^{img}, \mathbf{y}^{img})$ and $\mathcal{L}_{cls}(\mathbf{p}^{word}, \mathbf{y}^{word})$, namely

$$\mathcal{L}_{all} = \mathcal{L}_{cls}(\mathbf{p}^{img}, \mathbf{y}^{img}) + \mathcal{L}_{cls}(\mathbf{p}^{word}, \mathbf{y}^{word}). \tag{12}$$

Auxiliary Loss for Learning-based Strategy. Recall that in the learning-based strategy, to learn the codebook from gradients, we set the codebook as a trainable parameter and leverage the visual word representations to learn the image-level labels. Specifically, the visual word frequency $\mathbf{f}^{word}$ acquired in Eq. (5) is projected into the class probability space with an 1×1 *conv* layer. The predicted score is denoted by $\mathbf{p}^{w2i}$ so that the loss function is given as $\mathcal{L}_{cls}(\mathbf{p}^{w2i}, \mathbf{y}^{img})$.

We empirically found that learning with the loss $\mathcal{L}_{cls}(\mathbf{p}^{w2i}, \mathbf{y}^{img})$ solely tended to make the learned visual word representations in codebook $\mathbf{C}$ redundant. To tackle this problem, we add a regularization term to minimize the correlation between different latent visual word representations. Here, we use the DeCov loss (Cogswell et al., 2017), which reduces the correlations between rows in a matrix by minimizing their covariance, *i.e.*

$$\mathcal{L}_{decov} = \frac{1}{2}(\|\hat{\mathbf{C}}\|_F^2 - \|\mathrm{diag}(\hat{\mathbf{C}})\|_2^2), \tag{13}$$

where $\hat{\mathbf{C}}$ is the covariance matrix of $\mathbf{C}$, $\|\cdot\|_F$ denotes the Frobenius norm, and $\mathrm{diag}(\cdot)$ extracts the main diagonal elements of a matrix to a vector.

The auxiliary loss for the learning-based strategy is then given as the sum of aforementioned two losses

$$\mathcal{L}_{aux} = \mathcal{L}_{cls}(\mathbf{p}^{w2i}, \mathbf{y}^{img}) + \mathcal{L}_{decov}. \tag{14}$$

By regularizing Eq. (12) with Eq. (14), we present the overall loss function to optimize the network under the learning-based strategy

$$\mathcal{L}_{all} = \mathcal{L}_{cls}(\mathbf{p}^{img}, \mathbf{y}^{img}) + \mathcal{L}_{cls}(\mathbf{p}^{word}, \mathbf{y}^{word}) + \mathcal{L}_{aux}. \tag{15}$$

The overall training process with learning or memory-bank strategy is summarized in Alg. 1.

Algorithm 1: Training procedure of the proposed network.

Input: Image $\mathbf{I}$, label $\mathbf{y}^{img}$;
Params: Backbone network $\mathbb{E}(\cdot,\theta)$, codebook $\mathbf{C}$, hyper-parameters $\{k,\gamma,\tau,\rho\}$;
Initialize $\mathbb{E}(\cdot,\theta)$ and $\mathbf{C}$;
while *training* **do**
  Extract feature maps $\mathbf{F} = \mathbb{E}(\mathbf{I},\theta)$;
  Compute visual word label $\mathbf{y}^{word}$ via Eq. (2) to Eq. (4);
  Compute pooling features via Eq. (9) and Eq. (10);
  if *Learning-based strategy* **then**
    Compute visual word feature via Eq. (5);
    Compute loss via Eq. (15);
    Optimize $\mathbb{E}(\cdot,\theta)$ and $\mathbf{C}$;
  else *Memory-bank strategy*
    Compute loss via Eq. (12);
    Optimize $\mathbb{E}(\cdot,\theta)$;
    Reconstructing $\mathbf{C}'$ via Eq. (6);
    Update $\mathbf{C} \leftarrow \rho\mathbf{C}' + (1-\rho)\mathbf{C}$;
  end
end

4 Experiments

4.1 Implementation Details

Dataset. We evaluated our method on the PASCAL VOC 2012 dataset (Everingham et al., 2010) and MS COCO 2014 dataset (Lin et al., 2014). For all experiments, the mean Intersection-over-Union (mIoU) ratio was used as the evaluation criteria.

PASCAL VOC 2012 dataset (Everingham et al., 2010) consists of 21 semantic categories, including 20 foreground object classes and the background class. Following the common practice(Chen et al., 2017; Wang et al., 2020b; Chang et al., 2020b), this dataset is augmented with SBD dataset (Hariharan et al., 2011). The *train*, *val*, and *test* set of the augmented dataset consist of 10,582, 1449, and 1456 images, respectively.

MS COCO 2014 dataset (Lin et al., 2014) is a large-scale dataset with 81 semantic categories, including the background class. After excluding the images without annotations (Lee et al., 2021c), the MS COCO dataset consists of 82,081 and 40,137 images in *train* and *val* set, respectively.

Classification Network. For the network to produce CAMs, we used ResNet101 (He et al., 2016) pre-trained on ImageNet (Krizhevsky et al., 2012) as the backbone to extract convolutional feature maps. For PASCAL VOC and MS COCO datasets, the classification network was trained for 6 epochs, with a batch size of 16. To optimize the network, we used the SGD optimizer with momentum mechanism and set the momentum coefficient as 0.9. The learning rate was initially set to 0.01 for the backbone parameters and 0.1 for the other parameters. All learning rates were decayed every iteration with a polynomial decay scheduler. Specifically, in each iteration, the learning rate was multiplied by $(1-\frac{iter}{max\_iter})^{power}$, with $power = 0.9$.

Table 1: Evaluation and comparison of the generated CAMs and pseudo labels in mIoU. The best results are highlighted in **bold**.

Method	CAMs	+*CRF*	+*Ref.*
CAMs refined with PSA (Ahn and Kwak, 2018).			
PSA CVPR'2018	48.0	–	61.0
Mixup CAM BMVC'2020	50.1	–	61.9
SC-CAM CVPR'2020	50.9	55.3	63.4
SEAM CVPR'2020	55.4	56.8	63.6
PuzzleCAM arXiv'2021	51.5	–	64.7
AdvCAM CVPR'2021	55.6	62.1	68.0
CAMs refined with IRNet (Ahn et al., 2019).			
IRNet CVPR'2019	48.8	54.3	66.3
MBMNet ACM MM'2020	50.2	–	66.8
CDA arXiv'2021	50.8	–	67.7
VWE IJCAI'2021	55.1	60.9	69.5
CONTA NeurIPS'2020	56.2	–	67.9
AdvCAM CVPR'2021	55.6	62.1	69.9
Ours-M	56.9	62.6	71.1
Ours-L	**57.3**	**63.0**	**71.4**

As for the hyper-parameters, the number of visual words k and the weight factor γ in Eq. (10) were set to 256 and 2, respectively. The temperature parameter τ in Eq. (3) was empirically set to 1. For the memory-bank strategy, we had an extra momentum coefficient ρ in Eq. (7), which was set to 0.001. More details and the impacts of these hyper-parameters are reported in Section 4.5. Our code is available at https://github.com/rulixiang/vwe/tree/master/v2.

CAMs Refinement. The generated initial labels by directly segmenting CAMs with thresholds are usually very coarse (Ahn and Kwak, 2018; Ahn et al., 2019). To improve the quality of pseudo labels and the semantic segmentation performance, we adopted IRNet (Ahn et al., 2019) as the refinement approach for processing the initial coarse labels generated from the classification network. In practice, we used the official implementation[1] without changing their settings.

Segmentation Network. For the semantic segmentation network, we used the DeepLabV2 (Chen et al., 2017) system with ResNet101 (He et al., 2016) as backbone, which is a prevailing choice for WSSS task (Ke et al., 2021; Lee et al., 2021a; Chang et al., 2020b). For experiments on PASCAL VOC 2012 dataset (Everingham et al., 2010), we followed the default settings of DeepLabV2 (Chen et al., 2017), *i.e.*, the learning rate was initially set to 0.001 and decayed with a polynomial scheduler. The batch size and number of iterations were 10 and 20,000, respectively. We used a momentum optimizer with the momentum parameter of 0.9 and weights decay rate of 0.0005. For a fair comparison with other WSSS works, we evaluated the DeepLabV2 initialized with ImageNet (Krizhevsky et al., 2012) and MS COCO dataset (Lin et al., 2014) pre-trained weights. For experiments on the MS COCO dataset (Lin et al., 2014), we followed the same settings as the experiments on the PASCAL VOC dataset. The only difference was that we trained the segmentation network for 60,000 iterations since MS COCO consisted of much more samples.

[1] https://github.com/jiwoon-ahn/irn

Table 2: Semantic segmentation results on PASCAL VOC 2012 dataset. The best results are highlighted in **bold**. *Sup.* denotes supervision type. *Seg.* denotes segmentation network.

Method	*Sup.*	*Seg.*	*val*	*test*
Full Supervision.				
(1)† DeepLabV1† ICLR'2015	$\mathcal{F}$	–	75.5	–
(2) DeepLabV2 TPAMI'2017		–	76.3*	–
(2)† DeepLabV2† TPAMI'2017		–	77.6	79.7
(3) WideResNet38 PR'2019		–	80.8	82.5
(4) Res2Net101 TPAMI'2021		–	80.2	–
Image-level Supervision + Saliency Maps.				
OAA+ ICCV'2019	$\mathcal{I}+\mathcal{S}$	(1)†	65.2	66.4
Li *et al.* AAAI'2021		(2)	68.2	68.5
NSROM CVPR'2021		(2)	68.3	68.5
NSROM CVPR'2021		(2)†	70.4	70.2
DRS AAAI'2021		(2)†	70.4	70.7
EPS CVPR'2021		(2)†	**70.9**	**70.8**
AuxSegNet ICCV'2021		(3)	69.0	68.6
EDAM CVPR'2021		(2)†	**70.9**	70.6
Image-level Supervision Only.				
IAL IJCV'2020	$\mathcal{I}$	(2)	64.3	65.4
SEAM CVPR'2020		(3)	64.5	65.7
A^2GNN TPAMI'2021		(2)	66.8	67.4
VWE IJCAI'2021		(2)†	69.6	69.3
AdvCAM CVPR'2021		(2)	68.1	68.0
OC-CSE ICCV'2021		(3)	68.4	68.2
ESCNet ICCV'2021		(3)	66.6	67.6
CDA ICCV'2021		(3)	66.1	66.8
CPN ICCV'2021		(3)	67.8	68.5
PMM ICCV'2021		(4)	70.0	70.5
Ours-M		(2)	68.7	69.2[2]
		(2)†	**70.6**	70.4[3]
Ours-L		(2)	69.2	69.2[4]
		(2)†	**70.6**	**70.7**[5]

* *Accuracy obtained with our re-implementation.*
† *Backbone pre-trained on MS COCO dataset.*

4.2 Results on PASCAL VOC dataset

[2] http://host.robots.ox.ac.uk:8080/anonymous/XJDOJG.html
[3] http://host.robots.ox.ac.uk:8080/anonymous/JOOQBG.html
[4] http://host.robots.ox.ac.uk:8080/anonymous/YOXECB.html
[5] http://host.robots.ox.ac.uk:8080/anonymous/OQVYDO.html

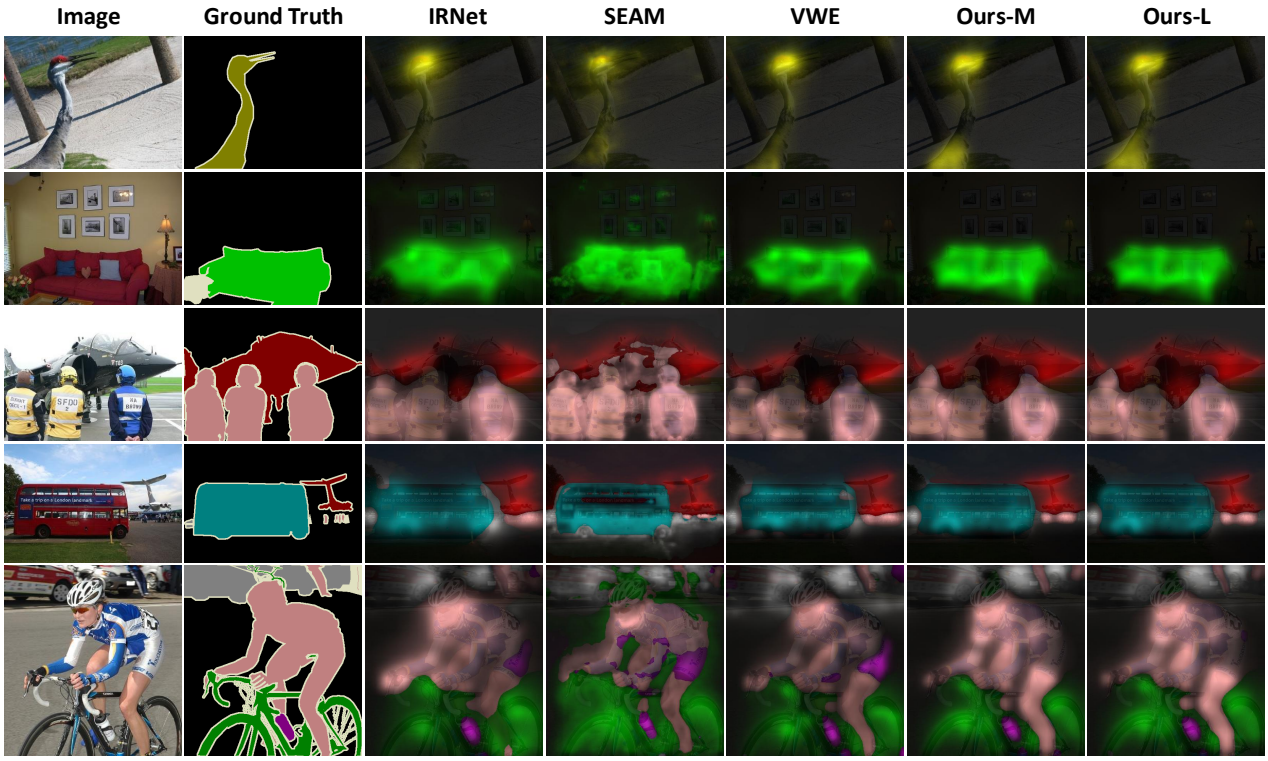


Fig. 6: Visualization of the generated CAMs. Different colors denote the activated regions of different semantic categories.

Table 1 reports the quantitative evaluation results of CAMs on the *train* set of the PASCAL VOC dataset. *CRF* denotes the generated CAMs are refined with dense CRF (Krähenbühl and Koltun, 2011). *Ref.* denotes the generated CAMs are refined with PSA (Ahn and Kwak, 2018) or IRNet (Ahn et al., 2019). The best results are highlighted in bold. We denote our methods with memory-bank strategy and learning-based strategy as Ours-M and Our-L, respectively. Ours results are compared with recent related works on improving the quality of CAMs, including AdvCAM(Lee et al., 2021a), SC-CAM (Chang et al., 2020b), CONTA (Zhang et al., 2020b), and SEAM (Wang et al., 2020b) *etc*. Table 1 shows that our methods with learning-based and memory-bank strategy could both remarkably surpass current state-of-the-art works. After further refinement with IRNet (Ahn et al., 2019), Our-M and Ours-L achieve 71.1% and 71.4% mIoU on the pseudo labels, respectively, which also outperform the competitors.

In Fig. 6, we visualize the generated CAMs and compare them with the results of recent methods, including IRNet (Ahn et al., 2019), SEAM (Wang et al., 2020b), and VWE (our previous work with HP and simple visual words encoder) (Ru et al., 2021). The results of the learning-based strategy (Ours-L) and the memory-bank strategy (Ours-M) are both presented. It is observed that our results typically activate more object regions and less mis-activated background, which is owing to that the proposed visual words learning module encourages to discover more objects, while HP aggregates local discriminative information and thereby reduces background regions. It is also noticed that the results of Ours-L and Ours-M are very close visually (Fig. 6) and numerically (Table 1), which indicates that both the learning-based and memory-bank strategy could work finely.

We use the refined CAMs as the pseudo labels to train regular semantic segmentation networks and compare the results on the *val* and *test* set of PASCAL VOC dataset. The results are reported in Table 2. For a fair comparison, we report the performance using DeepLabV2 with backbone pre-trained on ImageNet (Chen et al., 2017) and MS COCO (Lin et al., 2014). By default, the presented results are obtained with dense CRF post-processing (Krähenbühl and Koltun, 2011). It is observed that, for the WSSS methods with only image-level labels, our method obtains the best performance. Specifically, Ours-L achieves 69.2% and 70.6% mIoU on the PASCAL VOC *val* set with DeepLabV2 initialized with ImageNet and MS COCO pre-trained weights, respectively, which recover 90.7% and 91.0% of the upper bound of their fully-supervised counterparts. Our methods also achieve comparable performance with recent state-of-the-art WSSS methods us-

ing extra saliency maps, such as NSROM (Yao et al., 2021), DRS (Kim et al., 2021), EPS (Lee et al., 2021c), AuxSegNet (Xu et al., 2021), and EDAM (Wu et al., 2021). Our method also outperforms recent methods with superior backbone networks, such as PMM (Li et al., 2021), which uses Res2Net101 (Gao et al., 2021) as the backbone for semantic segmentation. Note that both Ours-M and Ours-L could surpass recent WSSS methods with only image-level supervision, which demonstrates the efficacy of our proposed learning-based and memory-bank strategies.

The qualitative results of our proposed method and some other methods' results, including LIID (Liu et al., 2020) and VWE (Ru et al., 2021), are presented in Fig. 7. We could observe that our method significantly outperforms other WSSS methods and coincides better with the ground-truth labels.

4.3 Results on MS COCO dataset

To further verify the efficacy of the proposed method, we conduct experiments on the MS COCO dataset (Lin et al., 2014), which consists of much more images and semantic categories than PASCAL VOC 2012 dataset. The quantitative evaluation results on the MS COCO dataset are presented in Table 3. We observe that Ours-L and Ours-M achieve 36.2% and 36.1% mIoU on the MS COCO *val* dataset, respectively. Both of them could outperform other WSSS methods with only image-level labels. Besides, our results are also better than the results of recent state-of-the-art WSSS methods with image-level labels and extra saliency cues. The superiority of the performance on the MS COCO dataset also demonstrates the efficacy of our methods.

We present some predicted example images of MS COCO *val* dataset in Fig. 8. It is observed that our method could produce comparable results with ground-truth labels, though the MS COCO dataset is much more challenging. However, when the background is complex, as presented in the last two columns, the predicted results are clearly worse than the ground-truth labels.

4.4 Ablation Study and Analysis

Quantitative Ablation Results. We conducted ablation experiments on the PASCAL VOC dataset to show the efficacy of the proposed methods. The quantitative evaluation results of the generated CAMs using baseline with different modules are presented in Table 4. VWE denotes the visual words learning module without DeCov regularization in our preliminary work (Ru et al., 2021). VWL-M and VWL-L denote the proposed visual words learning module with learning-based and memory-bank strategy, respectively. We observe that the both proposed HP and VWL module could improve the quality of the generated CAMs. Besides, our proposed memory-bank and learning-based strategy could further improve the mIoU on the *train* set to about 57%, which remarkably outperform recent state-of-the-art methods presented in Table 1.

Table 3: Semantic segmentation results on MS COCO dataset. The best results are highlighted in **bold**. *Sup.* denotes supervision type. *Seg.* denotes segmentation network.

Method	*Sup.*	*Seg.*	*val*
Image-level Supervision + Saliency Maps.			
DSRG CVPR'2018	$\mathcal{I}+\mathcal{S}$	DeepLabV2	26.0
Li *et al.* AAAI'2020		DeepLabV2	28.4
ADL TPAMI'2020		DeepLabV2	30.8
EPS CVPR'2021		DeepLabV2	35.7
AuxSegNet ICCV'2021		WideResNet38	33.9
Image-level Supervision Only.			
SEC ECCV'2016	$\mathcal{I}$	DeepLabV2	22.4
Saleh *et al.* TPAMI'2017		DeepLabV2	20.4
IAL IJCV'2020		DeepLabV2	27.7
SEAM CVPR'2020		WideResNet38	31.9
CONTA NeurIPS'2020		WideResNet38	32.8
CDA ICCV'2021		WideResNet38	33.2
PMM ICCV'2021		Res2Net101	35.7
Ours-M		DeepLabV2	36.1
Ours-L		DeepLabV2	**36.2**

Table 4: Ablation studies of our proposed methods on the *train* and *val* set. The best results are highlighted in **bold**.

Backbone	HP	VWE	VWL-M	VWL-L	*train*	*val*
ResNet50					48.3	47.0
ResNet101					49.5	48.4
ResNet101	✓				54.0 +4.5	53.1 +4.7
	✓	✓			55.1 +5.6	54.8 +6.4
	✓		✓		56.9 +7.4	56.4 +8.0
	✓			✓	**57.3** +7.8	**56.9** +8.5

Visual Ablation Results. Our intention of the proposed VWL and HP is to encourage the network to activate more object extents and fewer background regions, respectively. Though Table 4 shows that the proposed methods could improve the quality of CAMs, we still want to explore their effects on the generated CAMs. Therefore, we further visualize the CAMs generated by baseline, baseline with only HP, baseline with only VWL and our method. The visualization results are presented in Fig. 9. It is observed that VWL typically discovers more object extents, while both of them tend to activate adjacent background around objects. HP could remarkably alleviate this drawback since it

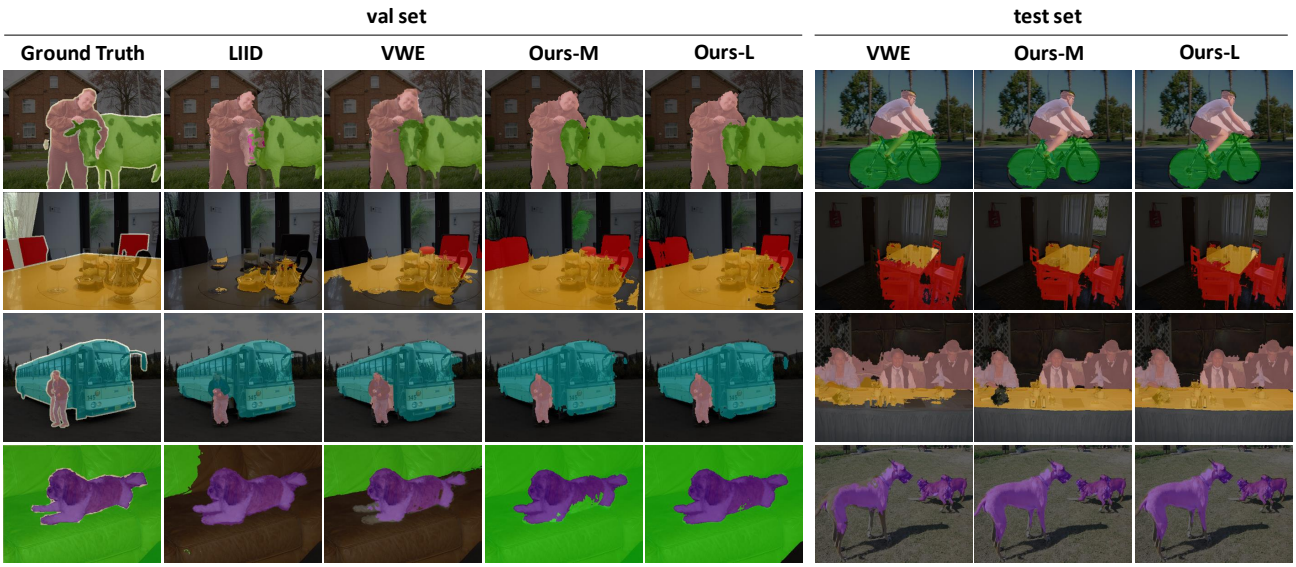


Fig. 7: Examples of the predicted segmentation from PASCAL VOC *val* and *test* set.

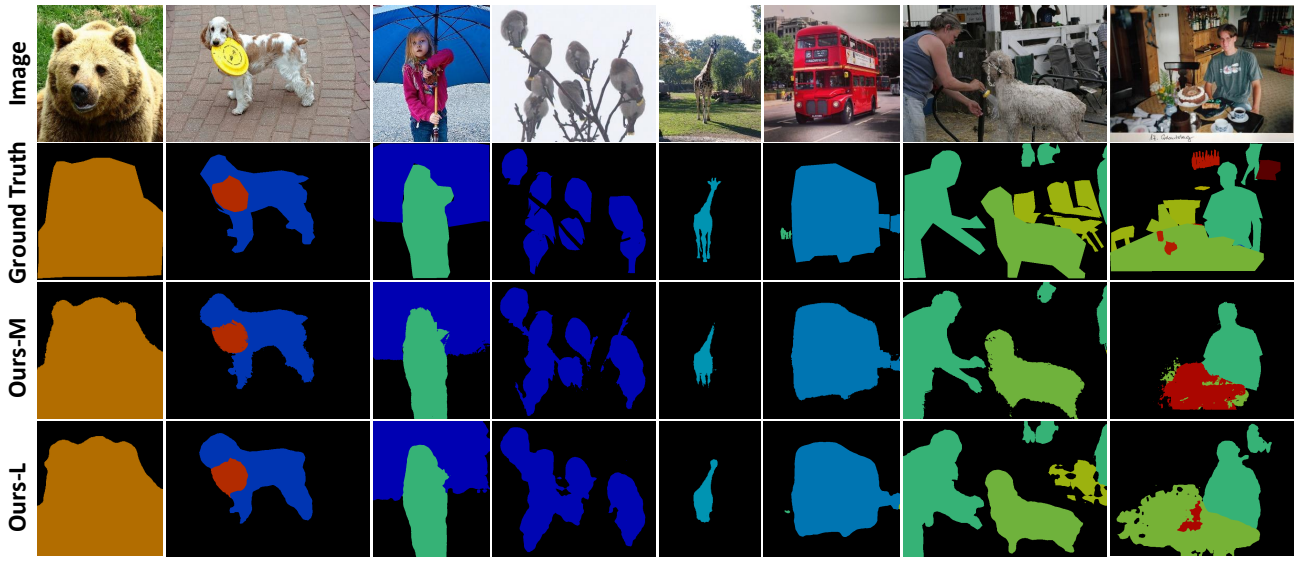


Fig. 8: Examples of the predicted labels from MS COCO *val* dataset.

aggregates local discriminative information instead of the whole image. Our method, which combines VWL and HP, could jointly mine more object regions and reduce the unexpected background. Fig. 9 also shows IRNet (Ahn et al., 2019) could further dampen the falsely activated regions and diffuse the object regions, so the CAMs can align better with the semantic boundaries.

Codebook Analysis. To verify whether the learning-based and memory-bank strategy could learn a reasonable codebook, we visualize the learned visual words represented in the codebook by extracting their corresponding regions in an image. The visualization results are presented in Fig. 10, where examples in each column are sampled from the images of a specific visual word. We show that both the learning-based strategy and memory-bank strategy could effectively learn visual word representations from images. For example, on the MS COCO dataset, our learning-based strategy could successfully decompose **person** to **face**, **body**, **hands** and **legs** *etc.*, which could be used to supervise the training of classification network and encourage more object extents to be discovered. Empirically, we also observe the learned visual words on MS COCO dataset typically consist of fewer noisy samples than the PASCAL VOC dataset, which indicates larger-scale dataset could benefit our visual words learning strategies.

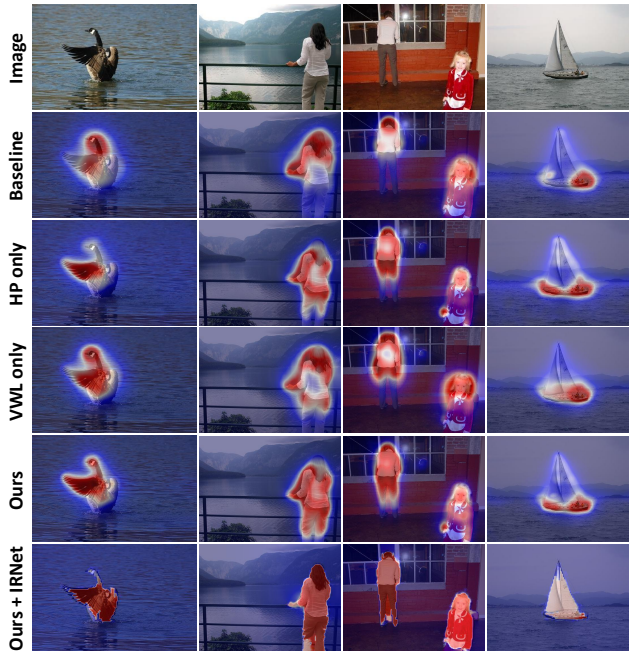


Fig. 9: Visualization of the generated CAMs.

Codebook Initialization. In this work, the codebook is simply randomly initialized. We do not use any extra pre-training or warm-up strategy for the codebook. To explore the impact of the initialization method, we present the performance of our method using random sample initialization (initializing the codebook with randomly sampled image features). The results are reported in Tab. 5. As shown in Tab. 5, the memory-bank strategy is not sensitive to the initialization method for the codebook, while the learning-based strategy achieves worse performance when using the random sample initialization. Technically, the codebook in the memory-bank strategy does not straightly affect the optimization process. Therefore, the impact of the initialization method is trivial. However, the codebook in the learning-based strategy is a trainable parameter and directly impacts the update process of the network parameters, thus notably affecting the performance of the generated CAM.

Table 5: The performance of the generated CAMs with different initialization methods. The results are evaluated on the PASCAL VOC *train* set.

	VWL-L	VWL-M
Random initialization	57.3	56.9
Random sample initialization	55.8	56.6

Learning-based versus Memory-bank. The learning-based strategy (VWL-L) and the memory-bank strategy (VWL-M) are both inspired by the simple Bag of Visual Words model. Specifically, in VWL-L, the visual word representation for an input image is automatically learned with the devised loss functions, while VWL-M extracts the visual word representations by online clustering. In other words, VWL-L and VWL-M model an image implicitly and explicitly, respectively. Therefore, we empirically find VWL-M could yield better visual words than VWL-L. Besides, as discussed in Section 4.4, compared to VWL-M, VWL-L is slightly sensitive to the initialization of the codebook. However, on the efficiency side, due to the online reconstruction operation, the training process of VWL-M takes a slightly longer time than VWL-L.

To better understand the quality of the generated visual words by the learning-based and memory-bank strategy, in Fig. 11, we visualize the extracted visual word features with t-sne (Van Der Maaten, 2014). The features for visualization are generated by averaging features of different visual word regions in each input image. As shown in Fig. 11, the clusters of VWL-L are more diverse than VWL-M's, *i.e.*, VWL-M learns better visual words than VWL-L, which is attributed to the explicit modeling of visual words in VWL-M. We then use the extracted visual word frequencies of each image to predict the image-level labels. The classification accuracies are reported in Tab. 6. VWL-M is still superior to VWL-L, demonstrating the better visual words learning capacity.

Table 6: The classification accuracies of the VWL-L and VWL-M. The performance is evaluated on the PASCAL VOC *train* set.

	VWL-L	VWL-M
Acc (%)	81.3	84.8

DeCov loss. In the loss function (Eq. (14)) for learning codebook in the learning-based strategy, we introduced the DeCov loss (Cogswell et al., 2017) to reduce the redundancy of the learned visual word representations. In Table 4, we show that learning visual words with DeCov loss could improve the mIoU of generated CAMs on PASCAL VOC *train* set from 55.1% to 57.3%. To further verify whether DeCov loss could eliminate the redundancy of the codebook, we visualized the similarity matrix of learned visual word representations. As presented in Fig. 12, when using DeCov loss regularization, the cosine similarity between two

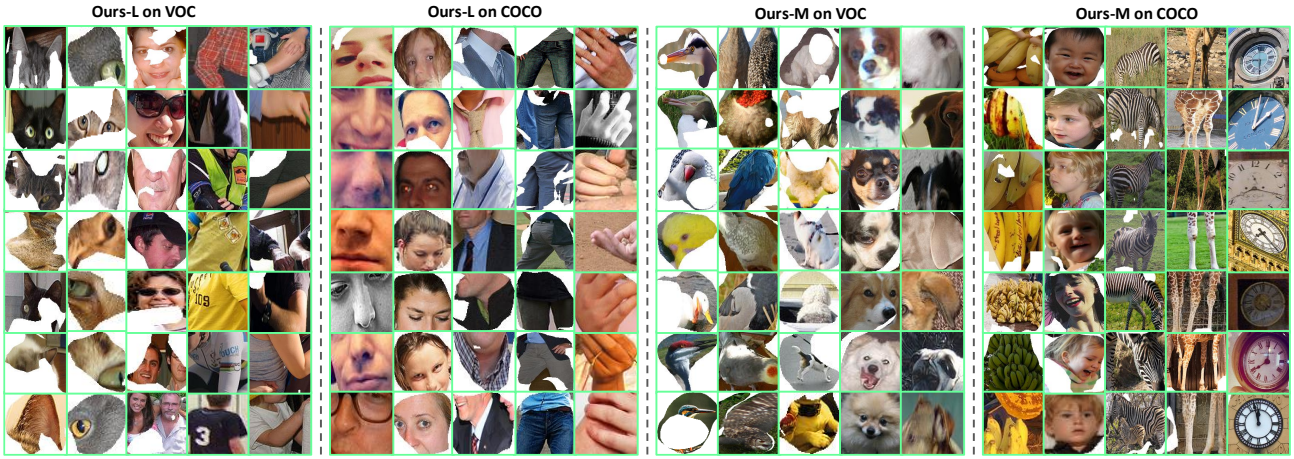
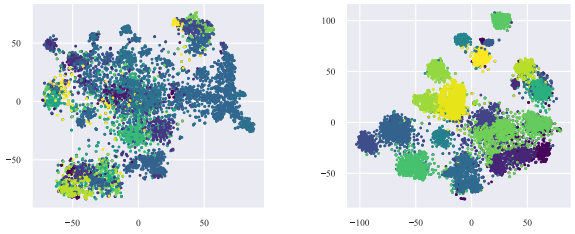


Fig. 10: Visualization of the visual words learned by learning strategy and memory-bank strategy. Each column denotes the example images sampled from a visual word category.



(a) Learning-based strategy (b) Memory-bank strategy

Fig. 11: t-sne visualization of the generated visual words. Different colors denote the different visual words

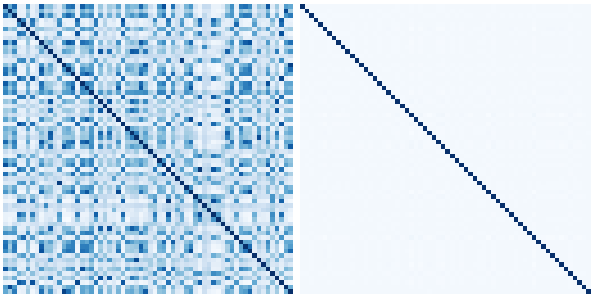


Fig. 12: Visualization of the similarity matrix between each visual words **without** (*left*) and **with** (*right*) DeCov loss.

different word representations is very close to 0. Taking the mIoU improvements in Table 4 into consideration, we demonstrate our regularization loss in Eq. (14) could successfully reduce codebook redundancy and improve CAMs quality.

GAP in HP for Object Completeness. Some previous works (Kolesnikov and Lampert, 2016; Zhou et al., 2016) show that GAP tends to overestimate the object size while GMP tends to underestimate it. In the design

	train	*val*
w/o GAP	41.1	39.9
w/ GAP	57.3	56.9

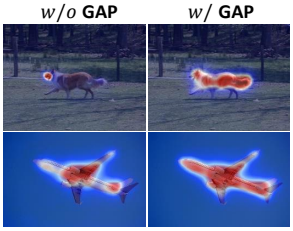


Fig. 13: Quantitative and visual results of hybrid pooling with (*w/*) and without (*w/o*) GAP.

of our hybrid pooling, we incorporate GAP to ensure object completeness. To verify the efficacy of GAP in HP, we present the quantitative and visual results of the generated CAMs using HP with and without GAP. As shown in Fig. 13, using GAP in HP could bring a large mIoU improvement of about 16%. The qualitative results also show that HP without GAP tends to discover only incomplete object regions while incorporating GAP could remarkably alleviate this problem.

Parallel Branches in HP. In the Hybrid Pooling module (HP), the parallel branches are used to produce multi-scale local information via max-pooling with different split sizes. Empirically, max-pooling with a small split size r aggregates less foreground and background information, leading to more discriminative object activation in CAMs. On the contrary, max-pooling with a large r aggregates more foreground and unexpected background information. In this work, we balance the background and foreground information aggregation by averaging the features with the different split sizes, *i.e.*, parallel branches in HP. We present the impact of the split sizes in Tab. 7. Tab. 7 shows that using max-pooling with the split size of $\{1, 2, 4\}$ in HP can activate

the background and foreground regions in CAMs well, and achieve the best performance.

Table 7: Impact of the set of split size in HP. The results are evaluated on the PASCAL VOC *train* set.

r	$\{1\}$	$\{1,2\}$	$\{1,2,4\}$	$\{1,2,4,8\}$
train	56.2	56.1	**57.3**	57.0

Comparison to GWRP and LSE. We compare the proposed HP with the global weighted rank pooling (GWRP) (Kolesnikov and Lampert, 2016) and Log-Sum-Exp pooling (LSE) (Pinheiro and Collobert, 2015). The experiments are conducted with ResNet101 (He et al., 2016) as the backbone (without visual word learning). As shown in Tab. 8, our method clearly outperforms GWRP and LSE. Besides, compared to GWRP and LSE, our HP is easier to implement since it only incorporates avg-pooling and max-pooling.

Table 8: Comparison of the pooling method on the PASCAL VOC *train* set.

	GAP	GWRP	LSE	Our HP
train	49.5	50.6	51.6	**54.0**

GAP in Visual Word Learning. As illustrated in Section 3.5, we use GAP instead our HP for predicting the visual word labels. We conduct experiments using HP in visual word learning to explore its impact. As shown in Tab. 9, using HP in visual word learning (VWL-HP) also achieves notable improvements, demonstrating the effectiveness of VWL. Nevertheless, using GAP in visual word learning (VWL-GAP) could further outperform VWL-HP. We analyze the reason in Section 3.5: The pseudo visual words are generated based on all pixels of the feature maps. HP mainly considers partial discriminative information while GAP could aggregate all information. Therefore, we think GAP is the better pooling method for predicting visual words, which is also verified in Tab. 9.

Table 9: Comparison of the pooling method in the visual word learning process on the PASCAL VOC *train* set.

	without VWL	VWL-HP	VWL-GAP
train	54.0	55.5	**57.3**

4.5 Effect of Hyper-parameters

This subsection presents the quantitative evaluation results of the generated CAMs on the PASCAL VOC *train* and *val* set with different hyper-parameter settings. All the results are evaluated and reported in mIoU.

Number of visual words k. In Table 10 (a), we present the impact of the number of visual words k by setting it to $\{128, 256, 384, 512\}$ and fixing other hyper-parameters (the classification network to generate CAMs is trained with the learning-based strategy). As observed in Table 10 (a), our method with different k could consistently outperform the baseline in Table 4, which demonstrates the effectiveness of our motivation. The best result is obtained with $k = 256$.

Weight factor γ. The effect of the weight factor γ in HP is presented in Table 10 (b), which is used to trade off the GAP and GMP features. We observe that $\gamma = 2$ works well, while the performance clearly decreases when $\gamma = 1, 4$ since the output feature will degrade to GMP or GAP when γ is too small or big.

Temperature parameter τ. In Eq. (3), we use a temperature parameter τ to control the smoothness of the visual word probabilities. As presented in Table 10 (c), we empirically observe that $\tau = 1.0, 0.8$ are proper values for generating CAMs with higher quality.

Momentum coefficient ρ. For the memory-bank strategy, a momentum coefficient ρ in Eq. (7) to manipulate the update rate of the codebook. In Table 10 (d), we show that $\rho = 1e^{-3}$ works finely since a large ρ makes the codebook dependent on the features from the current batch, while a smaller ρ means a slower update rate and may make the codebook not adaptable to training iterations.

5 Conclusion

Previous CAMs typically only cover partial discriminative object regions and some unexpected background. To tackle the first problem, we propose the visual words learning module. By enforcing the network to learn auxiliary visual words, more object regions could be activated. To perform unsupervised learning of visual words with only image-level labels, we devise the learning-based and memory-bank strategies to update the codebook. To mitigate the second problem, we propose hybrid pooling, which aggregates local maximum and global average features to simultaneously reduce background

Table 10: Impact of hyper-parameters.

	k			
	128	256	384	512
train	54.8	57.3	55.6	55.4
val	54.5	56.9	55.1	54.9

(a) Number of visual words

	γ			
	1	2	3	4
train	55.2	57.3	56.6	55.2
val	54.5	56.9	55.8	54.3

(b) Weight factor.

	τ			
	0.6	0.8	1.0	1.2
train	56.9	57.1	57.3	56.7
val	56.3	56.4	56.9	55.9

(c) Temperature parameter.

	ρ			
	$1e^{-4}$	$1e^{-3}$	$1e^{-2}$	$1e^{-1}$
train	55.6	56.9	56.3	55.9
val	55.0	56.4	55.6	55.3

(d) Momentum coefficient.

regions in CAMs and ensure object completeness. We experimentally demonstrated the superiority of our proposed method by surpassing recent state-of-the-art performance on the PASCAL VOC 2012 and MS COCO 2014 dataset.

References

Adams R, Bischof L (1994) Seeded region growing. IEEE Transactions on pattern analysis and machine intelligence 16(6):641–647

Ahn J, Kwak S (2018) Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 4981–4990

Ahn J, Cho S, Kwak S (2019) Weakly supervised learning of instance segmentation with inter-pixel relations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 2209–2218

Arandjelović R, Gronat P, Torii A, Pajdla T, Sivic J (2017) Netvlad: Cnn architecture for weakly supervised place recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 40(6):1437–1451

Araslanov N, Roth S (2020) Single-stage semantic segmentation from image labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 4253–4262

Badrinarayanan V, Kendall A, Cipolla R (2017) Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence 39(12):2481–2495

Bearman A, Russakovsky O, Ferrari V, Fei-Fei L (2016) What's the point: Semantic segmentation with point supervision. In: European conference on computer vision, Springer, pp 549–565

Chang YT, Wang Q, Hung WC, Piramuthu R, Tsai YH, Yang MH (2020a) Mixup-cam: Weakly-supervised semantic segmentation via uncertainty regularization. In: British Machine Vision Conference (BMVC)

Chang YT, Wang Q, Hung WC, Piramuthu R, Tsai YH, Yang MH (2020b) Weakly-supervised semantic segmentation via sub-category exploration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 8991–9000

Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2015) Semantic image segmentation with deep convolutional nets and fully connected crfs. In: International Conference on Learning Representations

Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2017) Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence 40(4):834–848

Cogswell M, Ahmed F, Girshick R, Zitnick L, Batra D (2017) Reducing overfitting in deep networks by decorrelating representations. In: International Conference on Learning Representations

Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, Franke U, Roth S, Schiele B (2016) The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3213–3223

Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. International journal of computer vision 88(2):303–338

Fan J, Zhang Z, Tan T, Song C, Xiao J (2020) Cian: Cross-image affinity net for weakly supervised semantic segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 34, pp 10762–10769

Gao SH, Cheng MM, Zhao K, Zhang XY, Yang MH, Torr P (2021) Res2net: A new multi-scale backbone architecture. IEEE Transactions on Pattern Analysis and Machine Intelligence 43(2):652–662

Gidaris S, Bursuc A, Komodakis N, Pérez P, Cord M (2020) Learning representations by predicting bags of visual words. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 6928–6938

Hariharan B, Arbeláez P, Bourdev L, Maji S, Malik J (2011) Semantic contours from inverse detectors. In:

2011 International Conference on Computer Vision, IEEE, pp 991–998

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778

Hou Q, Cheng MM, Hu X, Borji A, Tu Z, Torr PH (2017) Deeply supervised salient object detection with short connections. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3203–3212

Hou Q, Jiang P, Wei Y, Cheng MM (2018) Self-erasing network for integral object attention. Advances in Neural Information Processing Systems 31:549–559

Huang Z, Wang X, Wang J, Liu W, Wang J (2018) Weakly-supervised semantic segmentation network with deep seeded region growing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 7014–7023

Jiang PT, Hou Q, Cao Y, Cheng MM, Wei Y, Xiong HK (2019) Integral object mining via online attention accumulation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 2070–2079

Jo S, Yu IJ (2021) Puzzle-cam: Improved localization via matching partial and full features. In: 2021 IEEE International Conference on Image Processing (ICIP), pp 639–643

Ke TW, Hwang JJ, Yu SX (2021) Universal weakly supervised segmentation by pixel-to-segment contrastive learning. In: International Conference on Learning Representations

Kim B, Han S, Kim J (2021) Discriminative region suppression for weakly-supervised semantic segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 35, pp 1754–1761

Kolesnikov A, Lampert CH (2016) Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In: European conference on computer vision, Springer, pp 695–711

Krähenbühl P, Koltun V (2011) Efficient inference in fully connected crfs with gaussian edge potentials. Advances in neural information processing systems 24:109–117

Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems 25:1097–1105

Lee J, Kim E, Yoon S (2021a) Anti-adversarially manipulated attributions for weakly and semi-supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 4071–4080

Lee J, Yi J, Shin C, Yoon S (2021b) Bbam: Bounding box attribution map for weakly supervised semantic and instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 2643–2652

Lee S, Lee M, Lee J, Shim H (2021c) Railroad is not a train: Saliency as pseudo-pixel supervision for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 5495–5505

Li Y, Kuang Z, Liu L, Chen Y, Zhang W (2021) Pseudo-mask matters in weakly-supervised semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 6964–6973

Lin D, Dai J, Jia J, He K, Sun J (2016) Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3159–3167

Lin H, Upchurch P, Bala K (2019) Block annotation: Better image annotation with sub-image decomposition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)

Lin M, Chen Q, Yan S (2013) Network in network. arXiv preprint arXiv:13124400

Lin TY, 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

Liu L, Chen J, Fieguth P, Zhao G, Chellappa R, Pietikäinen M (2019) From bow to cnn: Two decades of texture representation for texture classification. International Journal of Computer Vision 127(1):74–109

Liu Y, Wu YH, Wen PS, Shi YJ, Qiu Y, Cheng MM (2020) Leveraging instance-, image-and dataset-level information for weakly supervised instance segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence

Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3431–3440

Oh Y, Kim B, Ham B (2021) Background-aware pooling and noise-aware loss for weakly-supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 6913–6922

Papandreou G, Chen LC, Murphy KP, Yuille AL (2015) Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In: Proceedings of the IEEE international conference on computer vision, pp 1742–1750

Passalis N, Tefas A (2017) Learning bag-of-features pooling for deep convolutional neural networks. In: 2017 IEEE International Conference on Computer Vision (ICCV), IEEE, pp 5766–5774

Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, et al. (2019) Pytorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems 32:8026–8037

Pinheiro PO, Collobert R (2015) From image-level to pixel-level labeling with convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1713–1721

Roy A, Todorovic S (2017) Combining bottom-up, top-down, and smoothness cues for weakly supervised image segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3529–3538

Ru L, Du B, Wu C (2021) Learning visual words for weakly-supervised semantic segmentation. In: International Joint Conference on Artificial Intelligence

Rubin DB (2019) Essential concepts of causal inference: a remarkable history and an intriguing future. Biostatistics & Epidemiology 3(1):140–155

Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G (2008) The graph neural network model. IEEE transactions on neural networks 20(1):61–80

Sculley D (2010) Web-scale k-means clustering. In: Proceedings of the 19th international conference on World wide web, pp 1177–1178

Song C, Huang Y, Ouyang W, Wang L (2019) Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 3136–3145

Sun G, Wang W, Dai J, Van Gool L (2020) Mining cross-image semantics for weakly supervised semantic segmentation. In: European Conference on Computer Vision, Springer, pp 347–365

Van Der Maaten L (2014) Accelerating t-sne using tree-based algorithms. The Journal of Machine Learning Research 15(1):3221–3245

Vernaza P, Chandraker M (2017) Learning random-walk label propagation for weakly-supervised semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7158–7166

Wang X, Liu S, Ma H, Yang MH (2020a) Weakly-supervised semantic segmentation by iterative affinity learning. International Journal of Computer Vision 128(6):1736–1749

Wang Y, Zhang J, Kan M, Shan S, Chen X (2020b) Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 12275–12284

Wei Y, Feng J, Liang X, Cheng MM, Zhao Y, Yan S (2017) Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1568–1576

Wu T, Huang J, Gao G, Wei X, Wei X, Luo X, Liu CH (2021) Embedded discriminative attention mechanism for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 16765–16774

Wu Z, Xiong Y, Yu SX, Lin D (2018) Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3733–3742

Xu L, Ouyang W, Bennamoun M, Boussaid F, Sohel F, Xu D (2021) Leveraging auxiliary tasks with affinity learning for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 6984–6993

Yao Y, Chen T, Xie GS, Zhang C, Shen F, Wu Q, Tang Z, Zhang J (2021) Non-salient region object mining for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 2623–2632

Zhang B, Xiao J, Wei Y, Sun M, Huang K (2020a) Reliability does matter: An end-to-end weakly supervised semantic segmentation approach. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 34, pp 12765–12772

Zhang D, Zhang H, Tang J, Hua XS, Sun Q (2020b) Causal intervention for weakly-supervised semantic segmentation. Advances in Neural Information Processing Systems 33

Zhang X, Wei Y, Feng J, Yang Y, Huang TS (2018) Adversarial complementary learning for weakly supervised object localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1325–1334

Zheng S, Jayasumana S, Romera-Paredes B, Vineet V, Su Z, Du D, Huang C, Torr PH (2015) Conditional random fields as recurrent neural networks. In: Proceedings of the IEEE international conference on computer vision, pp 1529–1537

Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2016) Learning deep features for discriminative lo-

calization. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2921–2929

Zhuang C, Zhai AL, Yamins D (2019) Local aggregation for unsupervised learning of visual embeddings. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 6002–6012