Learning Visual Words for Weakly-Supervised Semantic Segmentation

Lixiang Ru[1] , **Bo Du**[1*] and **Chen Wu**[2*]
[1]National Engineering Research Center for Multimedia Software, Institute of Artificial Intelligence,
School of Computer Science and Hubei Key Laboratory of Multimedia and Network Communication
Engineering, Wuhan University, Wuhan, China
[2]LIESMARS, Wuhan University, Wuhan, China
{rulixiang, dubo, chen.wu}@whu.edu.cn

Abstract

Current weakly-supervised semantic segmentation (WSSS) methods with image-level labels mainly adopt class activation maps (CAM) to generate the initial pseudo labels. However, CAM usually only identifies the most discriminative object extents, which is attributed to the fact that the network doesn't need to discover the integral object to recognize image-level labels. In this work, to tackle this problem, we proposed to simultaneously learn the image-level labels and local visual word labels. Specifically, in each forward propagation, the feature maps of the input image will be encoded to visual words with a learnable codebook. By enforcing the network to classify the encoded fine-grained visual words, the generated CAM could cover more semantic regions. Besides, we also proposed a hybrid spatial pyramid pooling module that could preserve local maximum and global average values of feature maps, so that more object details and less background were considered. Based on the proposed methods, we conducted experiments on the PASCAL VOC 2012 dataset. Our proposed method achieved 67.2% mIoU on the val set and 67.3% mIoU on the $test$ set, which outperformed recent state-of-the-art methods.

1 Introduction

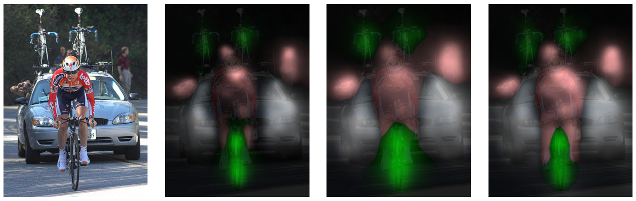
Semantic segmentation, aiming to assign a specific label for each pixel in an image, is a fundamental and hot topic in computer vision [Zhang *et al.*, 2019]. Usually, to train a semantic segmentation model with good performance, huge amount of images with pixel-level labels are indispensable. However, the annotation process of pixel-level labels is very expensive and time-consuming [Everingham *et al.*, 2015]. To alleviate this problem, In recent years, researchers have been dedicated to developing weakly-supervised semantic segmentation (WSSS) models which use weaker and cheaper labels, such as image-level labels [Ahn and Kwak, 2018; Wei *et al.*, 2018], point labels [Bearman *et al.*, 2016], bounding boxes [Wang *et al.*, 2019], and scribbles[Lin *et al.*, 2016].

Among them, semantic segmentation with image-level labels is the most challenging one. Existing methods typically follow a multi-step pipeline. They firstly train classification networks to generate the initial coarse pixel-level labels. The generated labels are then refined by methods such as Dense CRF [Chen *et al.*, 2017] and pixel affinity-based methods [Ahn and Kwak, 2018; Ahn *et al.*, 2019]. Based on the refined pseudo labels, a segmentation network will be trained as the final model for WSSS task using image-level labels.

The first step of the pipeline, *i.e.* generating the initial pseudo labels, is vital to the final semantic segmentation performance [Wang *et al.*, 2020; Chang *et al.*, 2020]. Prevailing image-level WSSS methods often train a classification network to produce class activation maps (CAM) [Zhou *et al.*, 2016] as the initial pseudo labels. However, CAM typically only identifies the most discriminative extents of a visual object. This problem is attributable to that the training process of network is guided by the classification loss which only aims to distinguish different classes. Since locating the most discriminative regions usually leads to better discriminability, there's no need for networks to discover the integral object. To tackle this problem, in this work, we proposed to simultaneously learn to classify the global image-level labels and local visual word labels. By enforcing the network to classify the global and local labels, more object extents could be discovered, so that the generated CAM could be more accurate. Since the visual word labels are not available in the image-level WSSS task, in each forward pass, they are generated in an unsupervised way. Concretely, the feature maps produced by CNN backbone are encoded by the $cosine$ similarities with each visual word in a trainable codebook.

Meanwhile, as shown in Fig 1, it's noted the choice of the feature aggregation layer also impacts largely on the quality of CAM. Empirically, the widely-used global average pooling (GAP) [Zhou *et al.*, 2016] often overestimates the object sizes and involves too much background since it averages all pixels in feature maps. On the contrary, global max pooling (GMP) [Oquab *et al.*, 2015], which only takes one pixel

[*]Corresponding Author. This work was supported in part by National Natural Science Foundation of China under Grant 61822113 and Grant 61971317, the National Key R&D Program of China under Grant 2018AAA0101100, the Natural Science Foundation of Hubei Province under Grant 2020CFB594, the Science and Technology Major Project of Hubei Province (Next-Generation AI Technologies) under Grant 2019AEA170, by the Wuhan Chang'e Information Technology Co., Ltd.



(a) Image (b) GMP (c) GAP (d) Ours

Figure 1: The generated CAM of our proposed method and baselines with GMP and GAP. Our results typically cover more object extents and less background.

in feature maps as output, usually underestimates the object sizes. To alleviate this problem, we proposed a new feature aggregation approach, named hybrid spatial pyramid pooling (HSPP), which incorporates both the global average and local maximums of feature maps as the output. As illustrated in Fig 3, In HSPP, the feature maps are firstly partitioned to multiple bins from coarse to fine levels as in spatial pyramid pooling [He *et al.*, 2015]. For bins in the same level, we pool them separately using GMP and average the aggregated features, so that only local maximums are involved. The features from different levels and the output feature of GAP are then averaged as the final output of HSPP. On this account, more discriminative object extents and fewer background regions are preserved in feature maps, which will improve the accuracy of the generated CAM.

To verify the effectiveness of our proposed approaches, we conducted extensive experiments on the frequently-used PASCAL VOC 2012 dataset [Everingham *et al.*, 2015]. The experimental results showed that our method could remarkably improve the performance of the generated CAM. After further refinement with IRNet [Ahn *et al.*, 2019] and training a DeepLabv2 segmentation network [Chen *et al.*, 2017] with the generated pseudo labels, we achieved 67.2% and 67.3% mIoU on the val and $test$ set, respectively, which surpassed the recent image-level WSSS methods.

The main contributions of this work are summarized as follows.

- We proposed to learn and classify the local visual word labels, which could enforce the network to discover more object extents and thus improve the quality of the generated pseudo pixel-level labels.
- We presented HSPP, a novel pooling method, which averaged the local maximum and global average features to alleviate the problem that the widely-used GAP and GMP can't estimate the objects accurately.
- We achieved 67.2% and 67.3% mIoU on the val and $test$ set of the PASCAL VOC 2012 dataset, which is the new state-of-the-art performance.

The rest of this paper is structured as follows. In Section 2, we'll introduce some related works within the context of our work. The proposed method is depicted in Section 3. Our experimental results are presented in Section 4.

2 Related Work

2.1 WSSS with Image-level Labels

WSSS with image-level labels is the most challenging one among all forms of supervisions. [Kolesnikov and Lampert, 2016] proposed the SEC principle to expand the initial seed cues to align the object boundaries. This framework is followed by many subsequent works [Huang *et al.*, 2018; Roy and Todorovic, 2017]. [Hou *et al.*, 2018; Wei *et al.*, 2017; Zhang *et al.*, 2018] explored the $erase$ strategy to erase the most discriminative region in each iteration so that more object extents could be discovered. [Ahn and Kwak, 2018] and [Ahn *et al.*, 2019] proposed pixel-level semantic affinity-based approaches with random walk inference [Vernaza and Chandraker, 2017] to refine the generated initial seed cues, which also achieved brilliant performance.

2.2 Generating Better Pseudo Labels

Recently, some works also dedicated to generating better initial pseudo labels. [Wei *et al.*, 2018] utilized the dilated convolution [Chen *et al.*, 2017] to enlarge the receptive field and discover more discriminative parts. $FickleNet$ randomly dropped the convolution kernels in each forward pass to enforce the network discover more object extents [Lee *et al.*, 2019]. Inspired by the fact that the segmentation masks should be scale-invariant, [Wang *et al.*, 2020] proposed to minimize the difference between the CAM of different scales. [Chang *et al.*, 2020] iteratively clusters the images to subclasses so that the CAM could activate more discriminative regions to differentiate different classes.

In this work, we also focus on semantic segmentation with image-level supervision and aim to improve the quality of initial pseudo labels.

3 Proposed Method

As illustrated in Fig 2, the proposed network for inferring CAM is mainly composed of a CNN backbone to extract convolutional feature maps, a Visual Word Encoder module (VWE) to encode local visual words and a hybrid spatial pyramid pooling (HSPP) layer to aggregated beneficial object information.

3.1 CNN Backbone

The CNN backbone in Fig 2 is composed of a sequence of convolutional layers and pooling layers. Let $\mathcal{X}$ be the set of N images in the $train$ set. The i-th image in $\mathcal{X}$, denoted as X, will be passed through the CNN backbone to obtain the convolutional feature map F with a spatial size of $h \times w$. Technically, any CNN architecture could be used as the backbone after removing its fully-connected layers. In this work, we used ResNet50 as the backbone network.

3.2 Visual Word Encoder

CAM guided by image-level labels often only covers the most discriminative extents of objects. The reason is that network doesn't need to discover the integral object to recognize different image classes. Our motivation is that if the network could be supervised with more fine-grained labels in the training procedure, it will be enforced to discover more semantic regions so that the generated CAM should be more accurate.

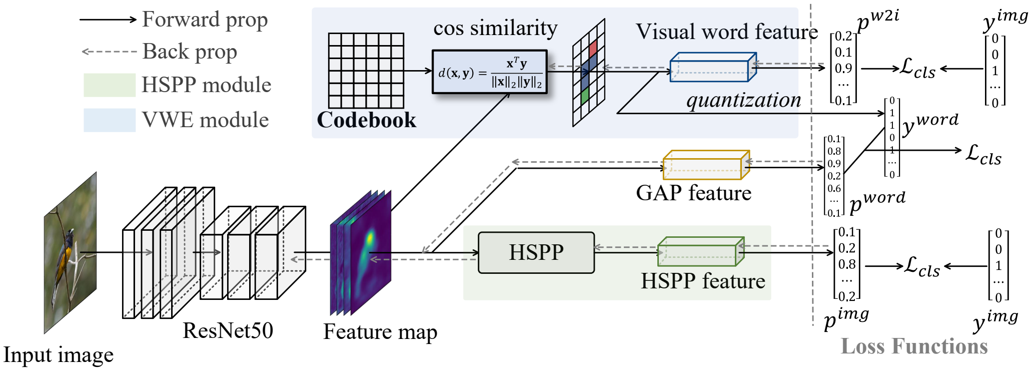


Figure 2: Overview of our proposed network. Our major contributions are the visual word encoder (VWE) and hybrid spatial pyramid pooling (HSPP) module.

To this end, we presented the visual words encoder for the WSSS task.

Since only image-level annotations are available in our task, to leverage local visual word labels to guide the training of networks, we designed an unsupervised visual word encoder (VWE). In the VWE module, a matrix $C \in \mathbb{R}^{k \times d}$ is defined as the codebook, where k is the number of words and d is the feature dimension. C is utilized to encode the extracted convolutional feature map $F \in \mathbb{R}^{h \times w \times d}$ to specific visual words. Here we use the cos distance to measure the similarity between the pixel at position i in F and the j-th word in C. The similarity matrix S is thus given as:

$$S_{ij} = \cos(F_i, C_j) = \frac{F_i^\top C_j}{||F_i||_2 ||C_j||_2}. \quad (1)$$

After obtained S, it will be normalized row-wise using $softmax$ function to compute the probability of the i-th pixel in F belonging to j-th word in codebook C.

$$P_{ij} = softmax(S_i) = \frac{\exp(S_{ij})}{\sum_{n=1}^{k} \exp(S_{in})}. \quad (2)$$

The visual word label of the Y_i for 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 P_{ij}, which is denoted as

$$Y_i = \arg\max_j P_{ij} \,. \quad (3)$$

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

In a BoVW model, the histogram distributions of each visual word are collected as the feature descriptor by counting their frequencies. 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 P. Therefore, the soft frequency assignment of the j-th word is

$$f_j^{word} = \frac{1}{hw} \sum_{i=1}^{hw} P_{ij}, \quad (4)$$

where f_j^{word} denotes the appearance frequency of the j-th word in F. As shown in Fig 2, f^{word} will be used to learn the image-level labels, *i.e.*, modelling the mapping relations between local visual words and image-level labels.

In a classic BoVW model, the codebook is usually identified as the clustering centroids of the feature representations extracted from all local visual words. However, in our model, the feature representations for visual words are online updated as the training procedure. Therefore, the codebook C should also be online updated. Following the approach in [Passalis and Tefas, 2017], in this work, the codebook C is set as a trainable parameter so that it could be learned automatically via the backpropagated gradients.

3.3 Hybrid Spatial Pyramid Pooling

To overwhelm the aforementioned disadvantages of GAP and GMP, in this work, we presented the hybrid spatial pyramid pooling (HSPP) which aggregates multi-scale local maximums and global averages of the convolutional map.

Consider the output feature map F with size of $h \times w \times d$ of the last convolutional layer, we first partition it to multi-scale divisions. As illustrated in Fig 3, each division with size of $\frac{h}{r} \times \frac{w}{r} \times d$ is pooled to a d-dimensional vector via $\mathbf{max}$ pooling, where $r \in \{1, 2, 4\}$ denotes the split size. F is thus aggregated to F^{max} with size of $r \times r \times d$. It's conspicuous that F^{max} only involves local maximum pixels so that less background is considered. We then pool F^{max} for the subsequent classification task. The pooled feature f_r^{max} with split size r is given by

$$f_r^{max} = \frac{1}{r^2} \sum_{i=1}^{r} \sum_{j=1}^{r} F_{i,j,:}^{max} \quad (5)$$

It's noted that operation in Eq 5 only preserves the maximum responses of local objects, which may corrupt the com-

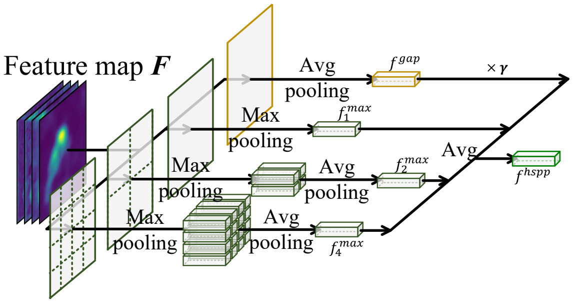


Figure 3: Illustration of the proposed HSPP.

pleteness of objects. To encourage the completeness of objects, in HSPP module, we also incorporate the results of GAP. Given the pooled feature of GAP layer, which is computed as

$$f^{gap} = \frac{1}{hw}\sum_{i=1}^{h}\sum_{j=1}^{w} F_{i,j,:}, \tag{6}$$

the final output of HSPP module is calculated by weighting the outputs in Eq 5 and Eq 6, computed as

$$f^{hspp} = \frac{1}{\gamma+3}\Big(\sum_{r\in\{1,2,4\}} f_r^{max} + \gamma f^{gap}\Big), \tag{7}$$

where γ is a weight factor that controls the proportion of the feature map of GAP. Leveraging Eq 7, more regions of foreground objects and less background are captured for classification, so that the generated CAM is more accurate.

3.4 Loss Function

Since only image-level annotations are available, the classification loss is indispensable to train the network. After obtaining f^{hspp} via Eq 7, the classification score for image label is computed with an additional 1×1 *conv* layer, denoted as $p^{img} = conv(f^{hspp}, W^{img})$, where W^{img} is the weight matrix of this layer. As a common practice, the multi-label soft margin loss [Paszke *et al.*, 2019] is employed to compute the classification loss

$$\begin{aligned}\mathcal{L}_{cls}(p^{img}, y^{img}) = \frac{1}{L}\sum_{i=1}^{L}[y_i^{img}\log\frac{\exp(p_i^{img})}{1+\exp(p_i^{img})} \\ + (1-y_i^{img})\log\frac{1}{1+\exp(p_i^{img})}],\end{aligned} \tag{8}$$

where 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 utilized to predict the visual word label y^{word} generated in previous steps. It's noted that y^{word} is generated based on all pixels in feature map F. Therefore, we use GAP here instead of HSPP to perform feature aggregation for predicting y^{word}. The predicted visual word score is thus denoted as $p^{word} = conv(f^{gap}, W^{word})$, where W^{word} is the weight matrix of the prediction layer. The classification loss for visual words is then denoted as $\mathcal{L}_{cls}(p^{word}, y^{word})$, which is in the same form as Eq 8.

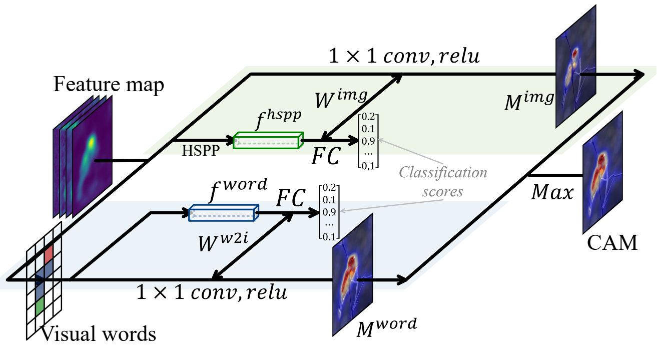


Figure 4: The procedure of CAM inference. The generated CAM is composed of 2 branches.

To model the mapping relations between visual words and image classes, the visual word frequency f^{word} acquired in Eq 4 is projected into the class probability space with an 1×1 *conv* layer with weight matrix W^{w2i}. The predicted score is denoted by p^{w2i}, such that the loss function is given as $\mathcal{L}_{cls}(p^{w2i}, y^{img})$.

The overall loss of the proposed network is finally formulated as the sum of the aforementioned loss terms.

$$\begin{aligned}\mathcal{L} = \mathcal{L}_{cls}(p^{img}, y^{img}) + \mathcal{L}_{cls}(p^{word}, y^{word}) \\ + \mathcal{L}_{cls}(p^{w2i}, y^{img}).\end{aligned} \tag{9}$$

3.5 CAM Inference

After trained the network with the loss function in Eq 9, the fixed parameters are used to infer CAM as the initial pseudo labels. As illustrated in Fig 4, the generated CAM is composed of 2 branches. In the top branch, we follow the original way in [Zhou *et al.*, 2016] to directly produce CAM with the feature map of the last *conv* layer and the weight matrix in prediction layer. Specifically, CAM for class c is given by weighting each feature map in F with its contribution to class c

$$M_c^{img} = \sum_{i=1}^{d}(W_{i,c}^{img} F_{:,:,i}). \tag{10}$$

M_c^{img} is further passed through a $relu$ layer to eliminate the negative values, which is denoted as $\hat{M}_c^{img}$. In the bottom branch, we use the encoded visual word maps and the learned weight matrix W^{w2i} to complement local information for original CAM.

$$M_c^{word} = \sum_{i=1}^{k}(W_{i,c}^{w2i} P_{:,:,i}). \tag{11}$$

M_c^{word} will also be passed through a $relu$ layer and gets $\hat{M}_c^{word}$. To capture more information, the final CAM is given by complementing the CAMs from two branches, which is denoted as

$$M_c = max(\hat{M}_c^{img}, \hat{M}_c^{word}). \tag{12}$$

The generated M_c will be used to produce the pseudo segmentation labels by segmenting the background and foreground with a background score threshold.

4 Experiments

4.1 Implementation Details

Dataset and Evaluation Criteria

The proposed network is trained and evaluated on the PASCAL VOC 2012 dataset [Everingham *et al.*, 2015]. This dataset includes 21 semantic categories, including 20 foreground classes and the background class. Following the common practice, this dataset is augmented with SBD dataset [Hariharan *et al.*, 2011]. The $train$ and val set of the augmented dataset consist of 10582 and 1449 images, respectively. For all experiments, the mean Intersection-over-Union (mIoU) ratio is used as the evaluation criteria.

Classification Network

As annotated in Fig 2, ResNet50 [He *et al.*, 2016] is employed as backbone to extract convolutional feature maps. The classification network is trained for 6 epochs, with batch size of 16. SGD optimizer is used during training. The initial learning rate is initially set to 0.01 for backbone parameters and 0.1 for the other parameters, The learning rate decays every iteration with a polynomial decay strategy. The number of visual words and the weight factor γ in Eq 7 are respectively set to 256 and 2. More details and the impacts of hyperparameters are reported in the supplementary material.

Refinement and Segmentation Network

To refine the generated initial pseudo labels, we adopted IRNet [Ahn *et al.*, 2019]. We used the official code released at GitHub without changing any settings [1]. The refined labels will be used to train a segmentation network. In our work, DeepLabv2 [Chen *et al.*, 2017] with ResNet101 [He *et al.*, 2016] as backbone is used as the segmentation network.

4.2 Ablation Study and Analysis

We first reported the ablation study results of our method in Table 1. Compared with the baseline, the proposed VWE module could bring an improvement of 2.8% on the $train$ set and 3.2% on the val set. The HSPP module also achieved 2.3% and 3.0% mIoU improvement on the $train$ and val set, respectively. After incorporating them together, the mIoU of the generated pseudo labels was further promoted to 52.9% on the $train$ set and 52.0% on the val set.

Baseline	VWE	HSPP	$train$	val
✓			48.3	47.0
✓	✓		51.1	50.2
✓		✓	50.6	50.0
✓	✓	✓	**52.9**	**52.0**

Table 1: Ablation studies of our proposed methods on the $train$ and val set. Baseline: ResNet50. VWE: Visual Word Encoder. HSPP: Hybrid Spatial Pyramid Pooling. The best results are highlighted in **bold**.

To verify whether the learned codebook could encode the input images reasonably, in each row of Fig 5, we visualized some samples of the learned visual words. The columns with green frames denoted the dominant samples in this category, while the last column with red frames presented some error samples. Fig 5 showed that the codebook could distinguish different visual words reasonably, which indicated the proposed VWE model works satisfactorily. It's also observed that different parts of a visual object could be effectively encoded. For example, the visual words in Row 2, Row 3, and Row 6 could be roughly interpreted as $head$, arm, and leg of $person$, respectively. With the supervision of the generated local visual words, the network could discover more object details, which is the source of the performance improvements in Table 1.

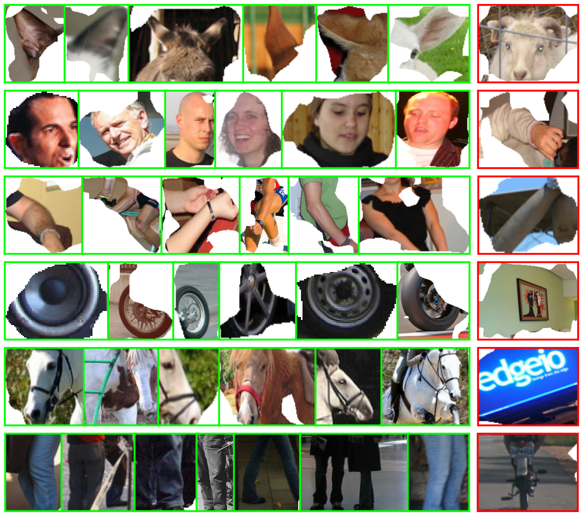


Figure 5: Samples of the learned words. In each row, images with green frame denote the dominant samples from this category, while images with red frame denote wrong words.

4.3 Comparison with State-of-the-art

In Table 2, we reported the mIoU of the generated initial and refined pseudo semantic segmentation labels and compared them with some recent approaches. Table 2 showed that, for the initial pseudo labels, our method remarkably outperformed the AffinityNet [Ahn and Kwak, 2018] and IRNet [Ahn *et al.*, 2019] baselines by ∼5% and SC-CAM [Chang *et al.*, 2020] by ∼2%. After further refinement using IRNet, our method still outperformed other recent methods on both the $train$ and val set.

Fig 6 showed the qualitative comparison between the generated CAM of classification network with GMP (the second row), GAP (the third row), and our proposed method using VWE and HSPP (the last row). Our method typically captured more regions of foreground objects and fewer background regions.

Based on the refined pseudo labels using IRNet, we trained a DeepLabv2 segmentation network with ResNet101 as the backbone and evaluated the results on the val and $test$ set. The evaluated results along with some other methods were presented in Table 3. Table 3 showed that our method achieved 67.2% mIoU on the val set and 67.3% mIoU on the

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

Method	Refinement	*train*	*val*
AffinityNet CVPR'2018		48.0	46.8
IRNet CVPR'2019		48.3	-
SC-CAM CVPR'2020	-	50.9	49.6
Ours		**52.9**	**52.0**
IRNet CVPR'2019		66.5	-
1Stage CVPR'2020	+ IRNet	66.9	65.3
Ours		**67.7**	**65.7**

Table 2: Evaluation and comparison of the generated pseudo labels in mIoU. The best results are highlighted in **bold**.

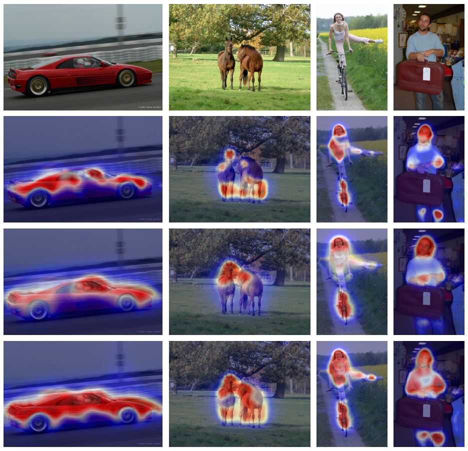


Figure 6: Visualization of the generated CAM. From top row to bottom row: input images, results of IRNet with GMP, results of IRNet with GAP, our results.

	Sup	Backbone	*val*	*test*
WideResNet38		WideResNet38	80.8	82.5
DeepLab	$\mathcal{F}$	VGG16	69.8	-
DeepLabv2		ResNet101	76.3	77.6
BoxSup ICCV'2015	$\mathcal{B}$	VGG16	50.7	51.7
BCM CVPR'2019		VGG16	66.8	-
ScribbleSup CVPR'2016	$\mathcal{S}$	VGG16	63.1	-
SEC ECCV'2016		VGG16	50.7	51.7
AffinityNet CVPR'2018		WideResNet38	61.7	63.7
DSRG CVPR'2018		ResNet101	61.4	63.2
IRNet CVPR'2019		ResNet50	63.5	64.8
SSDD ICCV'2019	$\mathcal{I}$	WideResNet38	64.9	65.5
SC-CAM CVPR'2020		ResNet101	66.1	65.9
SEAM CVPR'2020		WideResNet38	64.5	65.7
BES ECCV'2020		ResNet101	65.7	66.6
MCIS ECCV'2020		ResNet101	66.2	66.9
Ours w/o CRF	$\mathcal{I}$	ResNet101	66.3	66.3
Ours w/ CRF		ResNet101	**67.2**	**67.3**

Table 3: Evaluation of the semantic segmentation results in mIoU and comparison with other state-of-the-art methods. The best results are highlighted in **bold**. The supervision type (Sup) indicates: $\mathcal{F}$-Fully supervised, $\mathcal{B}$-Bounding box supervision, $\mathcal{S}$-Scribble supervision, $\mathcal{I}$-Image-level supervision.

$test$ set, which surpassed most recent other WSSS methods using image-level labels. Surprisingly, the proposed method also achieved better performance than recent methods with stronger supervision, such as bounding-box supervision and scribble supervision. The detailed mIoU results of each class on the val and $test$ set are available in the supplementary material.

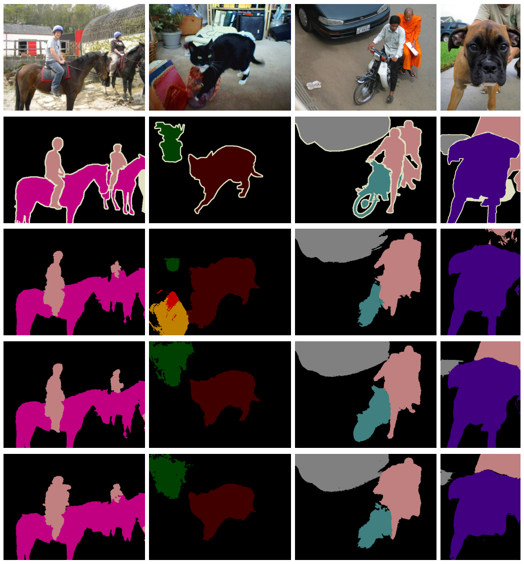


Figure 7: The predicted semantic segmentation masks of the PASCAL VOC val dataset. From top row to bottom row: input image, ground truth, results of LIID [Liu *et al.*, 2021], a *fully-supervised* DeepLabv2, and our *weakly-supervised* results.

In Fig 7, we presented the predicted semantic segmentation masks of the proposed method, a *fully-supervised* DeepLabv2 and LIID [Liu *et al.*, 2021], which's a recent WSSS work using image-level labels. Fig 7 showed that our method attained comparable performance with its *fully-supervised* counterpart, and both of them outperformed LIID . Our results are also very close to the ground truth labels.

5 Conclusion

In this work, we proposed an unsupervised visual word learning module to generate local visual word labels. By enforcing the classification network to learn the global image-level classes and the generated local labels, the generated CAM could discover more object extents. Meanwhile, we proposed a novel pooling approach that could preserve the local and global discriminative information and give more accurate estimations of objects. The experimental results demonstrated the effectiveness of our proposed methods and showed the proposed method achieved new state-of-the-art performance.

References

[Ahn and Kwak, 2018] 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.

[Ahn *et al.*, 2019] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. In *CVPR*, pages 2209–2218, 2019.

[Bearman *et al.*, 2016] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *ECCV*, pages 549–565. Springer, 2016.

[Chang *et al.*, 2020] Yu-Ting Chang, Qiaosong Wang, Wei-Chih Hung, Robinson Piramuthu, Yi-Hsuan Tsai, and Ming-Hsuan Yang. Weakly-supervised semantic segmentation via sub-category exploration. In *CVPR*, pages 8991–9000, 2020.

[Chen *et al.*, 2017] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE TPAMI*, 40(4):834–848, 2017.

[Everingham *et al.*, 2015] 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.

[Hariharan *et al.*, 2011] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, pages 991–998. IEEE, 2011.

[He *et al.*, 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE TPAMI*, 37(9):1904–1916, 2015.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[Hou *et al.*, 2018] Qibin Hou, PengTao Jiang, Yunchao Wei, and Ming-Ming Cheng. Self-erasing network for integral object attention. In *NeurIPS*, pages 549–559, 2018.

[Huang *et al.*, 2018] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *CVPR*, pages 7014–7023, 2018.

[Kolesnikov and Lampert, 2016] Alexander Kolesnikov and Christoph H Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *ECCV*, pages 695–711. Springer, 2016.

[Lee *et al.*, 2019] Jungbeom Lee, Eunji Kim, Sungmin Lee, Jangho Lee, and Sungroh Yoon. Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference. In *CVPR*, pages 5267–5276, 2019.

[Lin *et al.*, 2016] 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.

[Liu *et al.*, 2021] 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. *IEEE TPAMI*, 2021.

[Oquab *et al.*, 2015] Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *CVPR*, pages 685–694, 2015.

[Passalis and Tefas, 2017] Nikolaos Passalis and Anastasios Tefas. Learning bag-of-features pooling for deep convolutional neural networks. In *ICCV*, pages 5755–5763, 2017.

[Paszke *et al.*, 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8026–8037, 2019.

[Roy and Todorovic, 2017] Anirban Roy and Sinisa Todorovic. Combining bottom-up, top-down, and smoothness cues for weakly supervised image segmentation. In *CVPR*, pages 3529–3538, 2017.

[Vernaza and Chandraker, 2017] Paul Vernaza and Manmohan Chandraker. Learning random-walk label propagation for weakly-supervised semantic segmentation. In *CVPR*, pages 7158–7166, 2017.

[Wang *et al.*, 2019] Bin Wang, Guojun Qi, Sheng Tang, Tianzhu Zhang, Yunchao Wei, Linghui Li, and Yongdong Zhang. Boundary perception guidance: A scribble-supervised semantic segmentation approach. In *IJCAI*, pages 3663–3669, 2019.

[Wang *et al.*, 2020] 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.

[Wei *et al.*, 2017] 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.

[Wei *et al.*, 2018] 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.

[Zhang *et al.*, 2018] Xiaolin Zhang, Yunchao Wei, Jiashi Feng, Yi Yang, and Thomas S Huang. Adversarial complementary learning for weakly supervised object localization. In *CVPR*, pages 1325–1334, 2018.

[Zhang *et al.*, 2019] Qiming Zhang, Jing Zhang, Wei Liu, and Dacheng Tao. Category anchor-guided unsupervised domain adaptation for semantic segmentation. In *NeurIPS*, pages 433–443, 2019.

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