IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 September 2022, accepted 30 October 2022, date of publication 7 November 2022,
date of current version 7 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3220679

RESEARCH ARTICLE

Rethinking CAM in Weakly-Supervised Semantic Segmentation

YUQI SONG[1], XIAOJIE LI[1], CANGHONG SHI[2], SHIHAO FENG[3], XIN WANG[4], (Senior Member, IEEE), YONG LUO[5], AND XI WU[1]

[1]Department of Computer Science, Chengdu University of Information Technology, Chengdu 610225, China
[2]School of Computer and Software Engineering, Xihua University, Chengdu 610039, China
[3]Department of Computer Science, The University of Auckland, Auckland 1010, New Zealand
[4]University at Buffalo, SUNY, Buffalo, NY 14260, USA
[5]West China Hospital Sichuan University, Chengdu 610041, China

Yong Luo (e-mail: luoyonghx@163.com) and Xi Wu (e-mail: xi.wu@cuit.edu.cn)

Yuqi Song and Xiaojie Li have contributed equally to this work. This work was supported by National Key Research and Development Program of China (Grant No. 2020YFA0608000), the National Natural Science Foundation of China (Grant No. 42130608, 42075142) and the Sichuan Science and Technology program (Grant No. 23NSFSC2224, 2021YFQ0053, 2022YFG0152, 2020JDTD0020, 23ZHSF0169, 2022YFG0026, 2021YFG0018, 2020YJ0241).

ABSTRACT Weakly supervised semantic segmentation (WSSS) generally utilizes the Class Activation Map (CAM) to synthesize pseudo-labels. However, the current methods of obtaining CAM focus on salient features of a specific layer, resulting in highlighting the most discriminative regions and further leading to rough segmentation results for WSSS. In this paper, we rethink the potential of the ordinary classifier and find that if features of all the layers are applied, the classifier will obtain CAM with complete discriminative regions. Inspired by this, we propose Fully-CAM for WSSS, which can fully exploit the potential of the ordinary classifier and yield more accurate segmentation results. Precisely, Fully-CAM firstly weights feature with their corresponding gradients to yield CAMs of each layer, then fusing these layers' CAMs could generate an ultimate CAM with complete discriminative regions. Furthermore, Fully-CAM is encapsulated into a plug-in, which can be mounted on any trained ordinary classifier with convolution layer, and it exceeds its previous performance without extra training.

INDEX TERMS Weakly supervised semantic segmentation, class activation map, ordinary classifier, plug-in.

I. INTRODUCTION

Fully-supervised semantic segmentation (FSSS) [1], [2], [3], [4] aims to classify each pixel on image. With the development of deep learning, FSSS, as a basic computer vision task, has reached a major milestone. Unlike other general tasks such as object detection and classification, it is a data-driven task and requires the dense pixel-level masked label to train, but the cost of obtaining labels is huge. Object detection requires bounding box as supervision, and classification only requires category label as supervision. However, FSSS requires dense pixel-level annotation, the time cost of labeling pixel-level annotation is far higher than other tasks obviously. Therefore, much work has focused on weakly-supervised semantic segmentation (WSSS) in recent years. It is to synthesize pixel-level pseudo labels with low-level labels, such as scribble [5], [6], bounding box [7], [8], [9], points [10], [11] and image-level classification label [12], [13], [14], [15]. The image-level classification label is one of the most popular supervisions because it is straightforward to obtain. Simultaneously, it is also the most challenging for WSSS. The process of image-level WSSS methods is as follows: (1) the image-level classification label is used as the supervision to train a classifier which is usually a fully convolutional network (FCN) followed by a global average pooling (GAP) layer, and the features output by the last layer of the classifier is used as coarse localization named Class Activation Map (CAM) [16]; (2) refine the CAM to synthesize more accurate

The associate editor coordinating the review of this manuscript and approving it for publication was Byung Cheol Song.

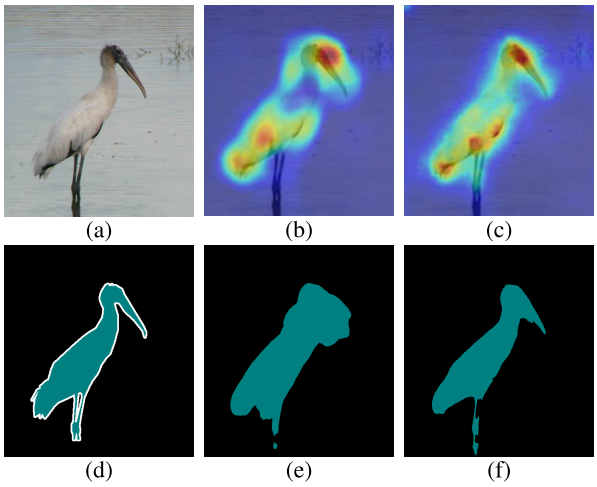


FIGURE 1. Quality of CAM and segmentation. (a) original image, (b) traditional CAM, which is generated by ordinary classifiers in the most weak semantic segmentation methods. (c) our CAM, which is generated by our method and can highlight whole objects. (d) ground truth, (e) Segmentation result by traditional CAM, (f) Segmentation result by our CAM.

pixel-level pseudo labels; (3) train a supervised semantic segmentation network with these pseudo labels and test its performance. To synthesize more accurate pixel-level pseudo labels based on CAM, DSRG [17] proposed using CAM as seed growth points and expansion. AffinityNet [13] proposed predicting the semantic similarity between adjacent coordinate pixel pairs in the image to diffuse CAM.

Generally speaking, high-quality CAM has a positive impact on the segmentation effect. as mentioned in many related works [18], [19], the quality of traditional CAM (as shown in Fig.1) is poor. It can only highlight the salient features of the object. Most previous works [18], [19], [20] on WSSS attributed the poor localization ability of CAM to the fact that the ordinary classifier can only highlight the most discriminative regions of each class. So, most of the works try to improve the performance of CAM by using complex training methods, e.g. Puzzle-CAM [19] proposed a separate and merged training method to narrow the gap between the global CAMs and local CAMs; SEAM [21] uses equivariant attention mechanism to fuse the CAMs from various transformed images and generate complete localization; CIAN [20] construct the affinity matrix of the two images by the self-attention mechanism and mine their common category location and uncommon category location; AdvErasing [22] gradually erases the most discriminator regions of CAM and guide the network to focus on other areas of the object. This not only increases the consumption of computing resources but also increases the difficulty of training. Nonetheless, we think they only see the appearance without deeper thinking, and the real reason for the poor localization ability of CAM is the insufficient utilization of information.

As far as we know, Convolutional Neural Network (CNN) based methods have different emphases on the features extracted from different layers. Generally, the object's low-level features (e.g., contour and texture features) are extracted in the shallow layer, and the high-level features (e.g., abstract features that are difficult to understand) are extracted in the deep layer. The CAMs of the ordinary classifier's layers also follow some rules. Specifically, the CAMs from the shallow layers of the classifier network have clear object contour, but with redundant noise, the CAMs from the deep layers relatively concentrate the object's discriminative regions, but the overall contour of the object has disappeared. Meanwhile, many works reuse features to improve their performance. For example, U-Net [4] integrates with some low-level and high-level features during upsampling and obtains a more accurate semantic segmentation effect. ResNet [23] reuses the previous extracted features by residual blocks to obtain higher classification accuracy. These works have shown that more features can be applied to overcome some limitations of traditional models. Therefore, more features could participate in the localizing object task in WSSS (see (i) in Fig.5), it highlights the whole area of the object by applying the features of previous layers.

In this paper, we rethink the potential of the ordinary classifier's CAM and find that the ordinary classifier already has sufficient capability to obtain CAM with more complete discriminative regions without complex training. To fully exploit the potential of the ordinary classifier, We propose a simple framework named Fully-CAM that applies the features from all convolution layers to gain CAM with complete discriminative regions for WSSS. The process of obtaining CAM can be divided into three steps: obtain the features of each convolution layer in the forward pass; obtain the gradients of each feature in the backward propagation; generate the ultimate CAM in the generation. Specially, In the backward propagation, we design the Computing Gradients Module (CGM) to obtain the gradients of all features at once. In the generation, we design the Fusing Localization Module (FLM) to generate the ultimate CAM by fusing all the features weighted by gradients. The main advantage of the proposed Fully-CAM is that it allows the classifier's all features to participate in localizing objects. As is known to all, the previous methods used a specific layer's features to determine the localization, which is regarded as the insufficient utilization of information and results in the absolute monopoly of the generated CAM over the localization task. However, our method allows all the features to participate in localizing and complementing each other's weaknesses with their strengths. It makes the ultimate CAM accurately localize objects. We also conduct extensive ablation studies and experimentally verify that the proposed Fully-CAM achieves additional performance.

Our main contributions are as follows:

- We experimentally verify that the ordinary classifier without complex training has enough capability to localize the whole object region.
- To make our method widely used, Fully-CAM is designed as a plug-in that can be mounted on any trained ordinary classifier with convolution layer without retraining and exceed their previous performances.

- We achieve the additional performance on the previous method in the WSSS through our CAMs on the PASCAL VOC 2012 val/test set with only the image-level classification labels.

II. RELATED WORK

Image-level weakly supervised semantic segmentation mainly studies two aspects: improving the quality of the CAM to highlight the whole discriminative regions of objects and synthesizing more accurate pixel-level pseudo labels. They are all inseparable from obtaining CAM. We first introduce the related progress on CAM and then related work in WSSS.

A. CLASS ACTIVATION MAP

CAM plays a significant role in interpreting CNN because it can visualize the basis of the model decision. At present, there are two mechanisms to obtain CAM. One is the traditional method [16] to obtain CAM by weighting the features based on the path weight of the full connection layer, and the other is Grad-CAM [24] to obtain CAM by weighting the features based on the gradients of backward propagation. The traditional method has strict requirements for the network structure of the classifier. The classifier must be a FCN followed by the GAP layer and the full connection layer. Sometimes the full connection layer can be removed, and the output result of the GAP can be directly used as the predicted confidence of each class. This strict constraint on the network results in that the traditional method can only obtain the last convolution layer's CAM. Later, the proposal of Grad-CAM makes it possible to obtain any layer's CAM in the network, and it can visualize CNN with any structure because Grad-CAM uses the gradient of backward propagation as the weight to weight the features to obtain CAM. Grad-CAM is flexible, but it lacks the importance of pixel space, result in the CAM is unclear. Note that the two visualization methods described above are only for a specific layer.

B. WEAKLY-SUPERVISED SEMANTIC SEGMENTATION

Compared with FSSS, WSSS uses low-level labels to generate pseudo pixel-labels to guide training, e.g., scribble [5], [6], bounding box [7], [8], points [10], [11] and image-level classification label [12], [13], [14], [15]. Most advanced methods utilize image-level labels to train models, and most of the works use the CAM obtained by the classifier to synthesize pseudo labels. DSRG [17] combines deep learning and seed region growing method, which uses CAM as seed growth points instead of manually selecting seed points to expand the entire region; AdvErasing [22] uses two classifiers, one to generate the CAM, and the other to iteratively erase the most discriminative areas in the CAM, and guide the network to focus on other areas of the object to highlight the entire area of the object; NL-CCAM [25] uses a linear function to calculate the coefficients of each CAM and weight the CAMs to make the foreground more prominent; AffinityNet [13] proposed predicting the semantic similarity between adjacent coordinate pixel pairs in the image to diffuse CAM; IRNet [12] generates a transition matrix from AffinityNet and extends the method to weakly supervised instance segmentation. There are also some advanced methods to use the attention mechanism to improve CAM on WSSS, e.g., CIAN [20] construct the affinity matrix of the two images by the self-attention and mine their common category location and uncommon category location. SEAM [21] proposed consistency regularization on predicted CAMs from various transformed images for self-supervision learning.

III. APPROACH

The overall pipeline of Fully-CAM is illustrated in Fig.2. Our framework consists of a training stage (not required) and a inferencing stage. In the training stage, we use the most common method to train a classification model to provide the basis for generating CAM in the inferencing stage. In the inferencing stage, there are three steps: forward pass, backward propagation, and generation to obtain CAM. The forward pass is used to obtain the feature of each convolution layer's output and predicted confidence score of the classifier, the backward propagation is used to obtain the gradient of each feature, and the generation is used to obtain ultimate CAM through features and gradients. In the backward propagation, we design the Computing Gradients Module (CGM) to obtain the gradients of a specific class. In the generation, Fusing Localization Module (FLM) is designed to generate ultimate CAM through the gradients and the features. It firstly generates the CAM of a single input image obtained by fusion of feature maps weighted by gradients. Then it fuses CAMs of different transformed images to generate the ultimate CAM.

A. TRAINING OF ORDINARY CLASSIFIER

Different from other WSSS methods, the classifier is applied the most ordinary classifier in our proposed method. In other words, it is applicable to any classifier with convolution layer.

We define I as input image and feature extraction as f. In previous methods in WSSS, the classification head often consists of a convolution layer $Conv$ with the number of output channels as number of class C and a global average pooling layer GAP. The advantage of classifier heads in previous WSSS methods is that they can obtain CAM more convenient, but it must to modify the trained classifier model structure and retrain, the confidence score y^{pred} is obtained by

$$y^{pred} = GAP(Conv(f(I))) \tag{1}$$

Nowadays, the classification head of most mature classifiers often consists of GAP layer and full connection layer FC, and the confidence score y^{pred} is obtained by

$$y^{pred} = FC(GAP(f(I))) \tag{2}$$

Since we classify multi-label data, the loss used is binary cross entropy loss (BCE), σ is $Sigmoid$, and loss is

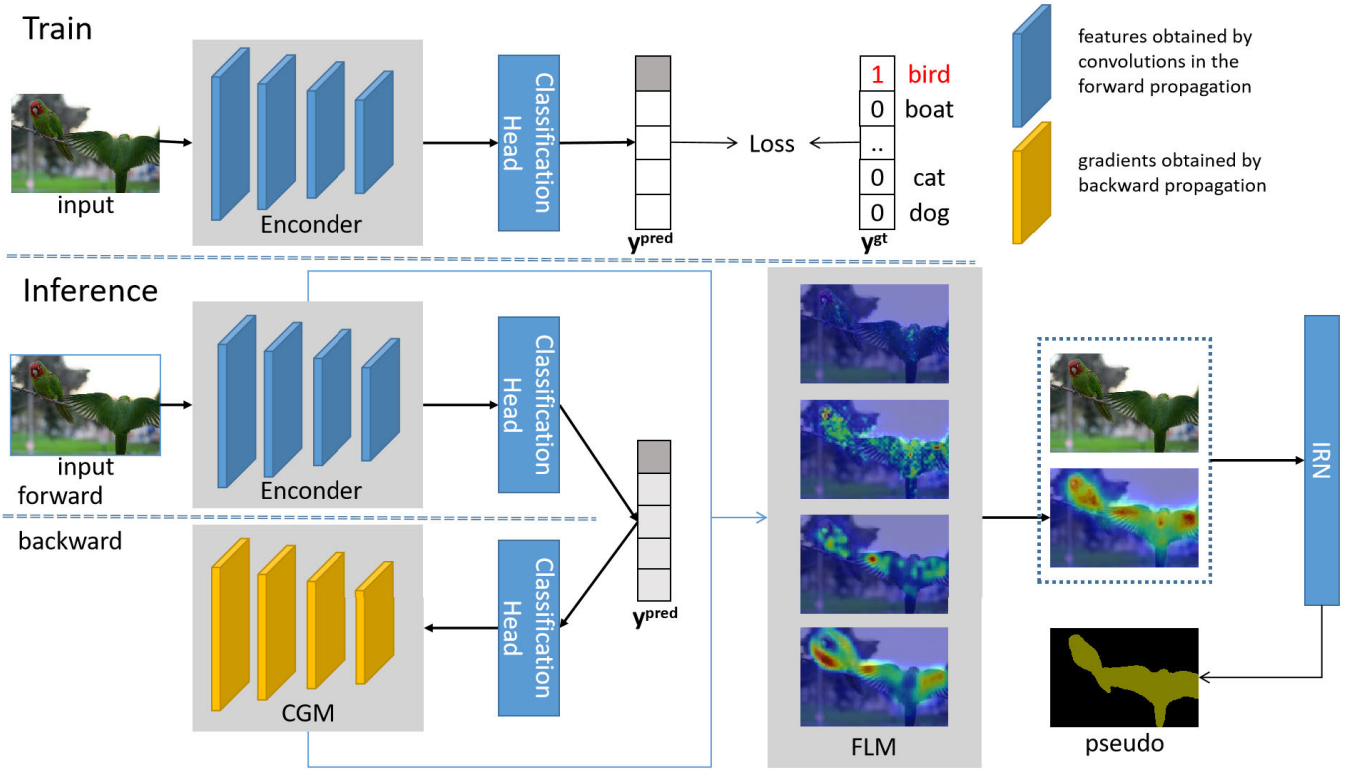


FIGURE 2. The architecture of our proposed Fully-CAM method. In the training stage(optional), we use the most common method to train a classification model. In the inference stage, Fully-CAM firstly saves the features extracted by the trained classifier in the forward propagation. Next, these features are given into the computing gradients module (CGM) to compute the gradient of the specified class in the backward propagation. Finally, all extracted features and corresponding gradients are fused by the fusing localization module (FLM) to generate the ultimate CAM. IRN [12] provide service of generating pseudo pixel-label for our high-quality CAM.

obtained by

$$loss(y^{pred}, y^{gt}) = -\frac{1}{C}\sum_{i}(y^{gt}[i] * log(\sigma(y^{pred}[i])) + (1 - y^{gt}[i]) * log(\frac{e^{-y^{pred}[i]}}{\sigma(y^{pred}[i])})) \quad (3)$$

In our method, we generalize the classification head to make it universal and can be used directly without modification.

B. COMPUTING GRADIENTS MODULE

We all know that there may be multiple classes of objects on an image, and we need to distinguish the discriminative regions of different classes. This section will introduce in detail how CGM obtains the gradients of a layer's feature map of a specified class. Fig.3 shows the process of CGM.

Formally, let *classifier* denote the image classifier and θ represent its parameters. For a given image I, when inputting I to the classifier, we can obtain the predicted score under a specific class c_i defined as

$$y_{c_i}^{pred} = P(y_{c_i}|I, \theta) = classifier(I, \theta) \quad (4)$$

Let A^n be the output feature maps of the n-th convolution layer in the network, the shape of A^n is $(1, K, W, H)$. A^{nk} $(k \in [1, K])$ is the k-th feature map within A^n and its shape is (W, H). The gradients of the prediction score y^{pred} under a

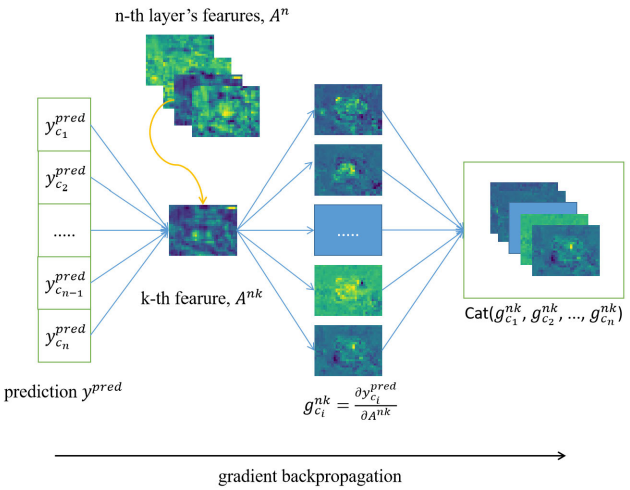


FIGURE 3. The schematic of obtaining gradients for the specific feature. The predicted scores of each class compute the gradient of the feature A^{nk}, and finally, these gradients are concatenated to form the gradient of all classes to the feature map A^{nk}. Cat: Concatenate.

specific class $c_i \in C$ in the feature map A^{nk} can be obtained by

$$g_{c_i}^{nk} = \frac{\partial y^{pred}}{\partial A^{nk}}, \quad c_i \in [c_1, c_2, \ldots, c_n]$$
$$g^{nk} = Cat(g_{c_1}^{nk}, g_{c_2}^{nk}, \ldots, g_{c_n}^{nk}) \quad (5)$$

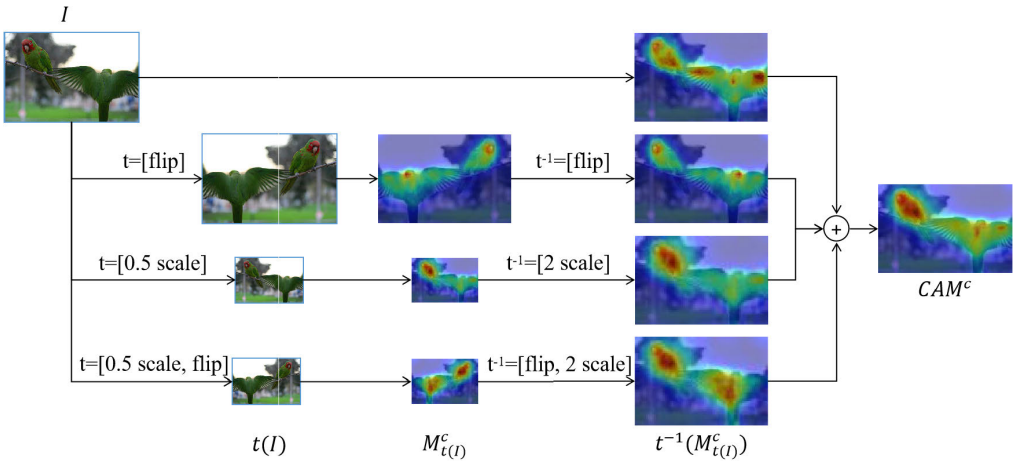


FIGURE 4. The fusion of the CAMs of different transformation on an image. I: Original Image. t: Function of transformation for original image. t^{-1}: Function of inverse transformation of t for CAM of $t(I)$. CAM: ultimate CAM.

The g^{nk} represents the gradients of the all classes C. Note that g^{nk} is a three-dimensional matrix of shape $C \times W \times H$, since the gradient of back propagation is computed for each predicted class $y_{c_i}^{pred}$, the number of channels in g^{nk} is C. Next, the gradients of a specific class is what we need, we have to filter the gradients. Let c denote the c-th class in the ground truth y^{gt} of image I, y^c is a vector of shape $1 \times C$ that is used as a label for the target class c by one-hot encoding. The gradients $g^{nkc}(1 \times W \times H)$ of the target category c in the feature map A^{nk} can be obtained by

$$g^{nkc} = y^c \cdot g^{nk} \tag{6}$$

Due to the shape of g^{nkc} should be the same as A^{nk}, we need to squeeze the g^{nkc}, and $\hat{g}^{nkc}$ of shape $W \times H$ can be obtained by:

$$\hat{g}^{nkc} = squeeze(g^{nkc}, 0) \tag{7}$$

C. FUSING LOCALIZATION MODULE

The Fusing Localization Module is designed to generate a CAM with complete discriminative regions by fusing various CAMs from different convolution layers and the different transformed input images.

Before fusion, we first introduce how to generate the CAM of a certain convolution layer of a specific class. To obtain the CAM for the n-th convolutional layer in CNN, it first multiplies the activation value of each location in the feature map by a gradient as the weight to obtain the CAM of the k-th feature map of the n-th convolution layer CAM^{nkc} and (i, j) represents the spatial location, and the result is obtained by:

$$CAM_{ij}^{nkc} = relu(A_{ij}^{nk}) \cdot relu(\hat{g}_{ij}^{nkc}) \tag{8}$$

g_{ij}^{nkc} indicates the influence of target category c to A_{ij}^{nk}. If the gradient is negative, it is irrelevant to A_{ij}^{nk}. Similarly, A_{ij}^{nk} also may be negative, and it is regarded as information redundancy. Moreover, there will be a lot of floating-point operations, and we set all negative values to zero for ease of computation. We have obtained the CAM of the k-th feature map of the n-th convolution layer, but we cannot fuse it directly. The reason is that there is a huge numerical gap between values of CAM of each feature map of each convolution layer. If it is accumulated simply, it will make the CAM with a large value play an absolutely dominant role. In order to make each CAM reflect its characteristics, we normalize each CAM so that the value range is between [0, 1]. Then, the normalized CAM^{nkc} are linearly combined along the channel dimension to obtain the CAM CAM^{nc}, which is formulated as follows:

$$CAM^{nc} = \sum_k \frac{CAM^{nkc}}{max(CAM^{nkc})} \tag{9}$$

We can get the CAMs of all convolution layers through the above steps. However, due to the size, stride, and padding of the convolution kernel and downsampling in the network, the obtained CAMs are different in size. As shown in the inference of Fig.2, due to the features in different layers with different sizes, We need to restore the CAMs to the image I size through linear interpolation. the restored CAM of the n-th convolution layer is obtained by

$$\hat{CAM}^{nc} = interpolate(CAM^{nc}, size(I)) \tag{10}$$

Finally, CAMs from all convolution layers are fused to generate the ultimate CAM of a specific class of image by:

$$CAM^c = normalize(\sum_n \hat{CAM}^{nc}) \tag{11}$$

where CAM^c is the ultimate CAM and $\hat{CAM}^{nc}$ represent the CAM from n-th convolution layers. From (4) to (11), all the features from all the convolution layers produce the ultimate CAM. Different from previous approaches (such as traditional CAM [16], Grad-CAM [24]), whether a certain location of the image is highlighted and its degree of highlighting is not determined by one or several features but by

all the features captured by the network. The Fully-CAM method, which uses all the features captured by the network, can achieve more accurate and fine localization than other methods.

Although Fully-CAM can capture accurate and detailed location information, we used a little trick to further improve the highlight localization performance. As shown in Fig.4, we send the original and transformed images to the network to get the corresponding CAMs and integrate information from both. For example, it uses the flipped image to get the CAM of the flipped image, and it is necessary to flip the CAM of the flipped image back for matching the CAM of the original image, then fuse the CAMs of the original image and flipped image. Here we denote the scaling, flipping, and other transformations as t, the inverse transformations as t^{-1}, and the process of formula 1-8 as τ. Therefore, we can get the enhanced ultimate CAM by

$$\begin{aligned} CAM &= \tau(I) \\ CAM^{t_1} &= \tau(t_1(I)) \\ CAM^{t_2} &= \tau(t_2(I)) \\ &\cdots \\ CAM^{t_n} &= \tau(t_n(I)) \\ \hat{CAM} &= CAM + \sum_{i}^{n} t_i^{-1}(CAM_{t_i}) \end{aligned} \tag{12}$$

By (12), we get the CAM of the transformed image by $t(I)$, then inversely transform the CAM by $T_{-1}(CAM)$, and exchange information with other CAMs that are inversely transformed to obtain the enhanced ultimate CAM $\hat{CAM}$. In this way, the information can be utilized to the greatest extent, useless information can be filtered out, and the accuracy of localizing can be improved.

IV. EXPERIMENTAL RESULTS

A. DATASET & IMPLEMENTATION DETAILS

1) DATASET

PASCAL VOC 2012 dataset [26] is used in our experiments which is the most representative dataset in WSSS. It includes 4369 images, 1,464 images for training, of witch 1,449 for validation and 1,456 for testing. Note that, to be consistent with the experience of previous works [13], [19], [27], [28], [21], we also introduce Semantic Boundary Dataset [29] as an augmented training set with 10,582 images. Mean Intersection-over-Union (mIoU) is used to measure the performance of different methods.

2) IMPLEMENTATION DETAILS

Our experiments are implemented based on PyTorch 1.10 with ResNet-50 and ResNet-101 as the backbone network for WSSS. We follow the previous work [19] to set the parameters of the experiment. Specifically, we use Adam optimizer with an initial learning rate of 0.1, weight decay of 0.0005, $\alpha = 4$ as the maximum, and linearly ramped up α to its maximum value by half epochs. The batch size is 32 with 15 epochs on four NVIDIA 3080 GPUs for training the classifier. The batch size is 24 with three epochs on three NVIDIA 3080 GPUs for training IRNet [12]. The batch size is 24 with 50 epochs, and the initial learning rate is 0.007 on four NVIDIA 3080 GPUs for training DeepLab. For data augmentation, we first randomly resize the image to 320×640 and randomly flip it, and the crop size is 512. In the inference stage, we randomly flip the image and use multi-scale (the scale ratio is set to {0.5, 1, 1.5, 2}) on a single 3080 GPU.

B. ABLATION STUDIES

Our Fully-CAM has three essential aspects: (1) the CAMs of all features are weighted and fused; (2) the CAMs of different transformations are fused; (3) Fully-CAM is plug and play, which improves the performance of any trained ordinary classifier with a global average pooling layer without extra training. We perform experiments to study the effect of different aspects of our model.

Previous work only used the features of a specific layer as the basis for CAM, which is a behavior of insufficient information utilization. Thus, we first validate the influence of applying the features of different layers (see Table.2). Due to too many convolution layers in the network, if each feature of the convolution layers needs to test the performance separately, the workload will be huge. Therefore, we artificially selected several representative convolution layers in the network and finally applied all features of the network. It is easy to see that when more and more convolution layers' features are applied, the localizing effect of CAM is better and better with mIoU(%) from 44.76% to 53.88%. This further proves the importance of features for CAM.

Furthermore, for three ways to obtain CAM: traditional CAM [16], Grad-CAM [24], and Grad-CAM++ [35], they have their advantages. Traditional CAM can only obtain the localization of the last layer's feature maps, Grad-CAM and Grad-CAM++ can obtain the localization of any layer's feature maps, but the noise will affect their localizing. The common point is to obtain localization only from the feature maps of a specific layer. As we said above, this is a manifestation of insufficient use of information. As shown in Table.3, if localization is obtained only from a specific layer's feature maps, the localizing effect will be challenging to improve. In contrast, our method has better results.

We believe that the classifier pays attention to different regions for different transformed input images, and we use the data enhancement strategy of random size and random flip when training the classifier. Therefore, we have done ablation experiments on multi-scale and flipping in the inference step. Table.4 shows the effectiveness of our introduced transformations. It is easy to see that with the increase in the number of transformed images, the mIoU of CAM is from 51.28% to 53.88%.

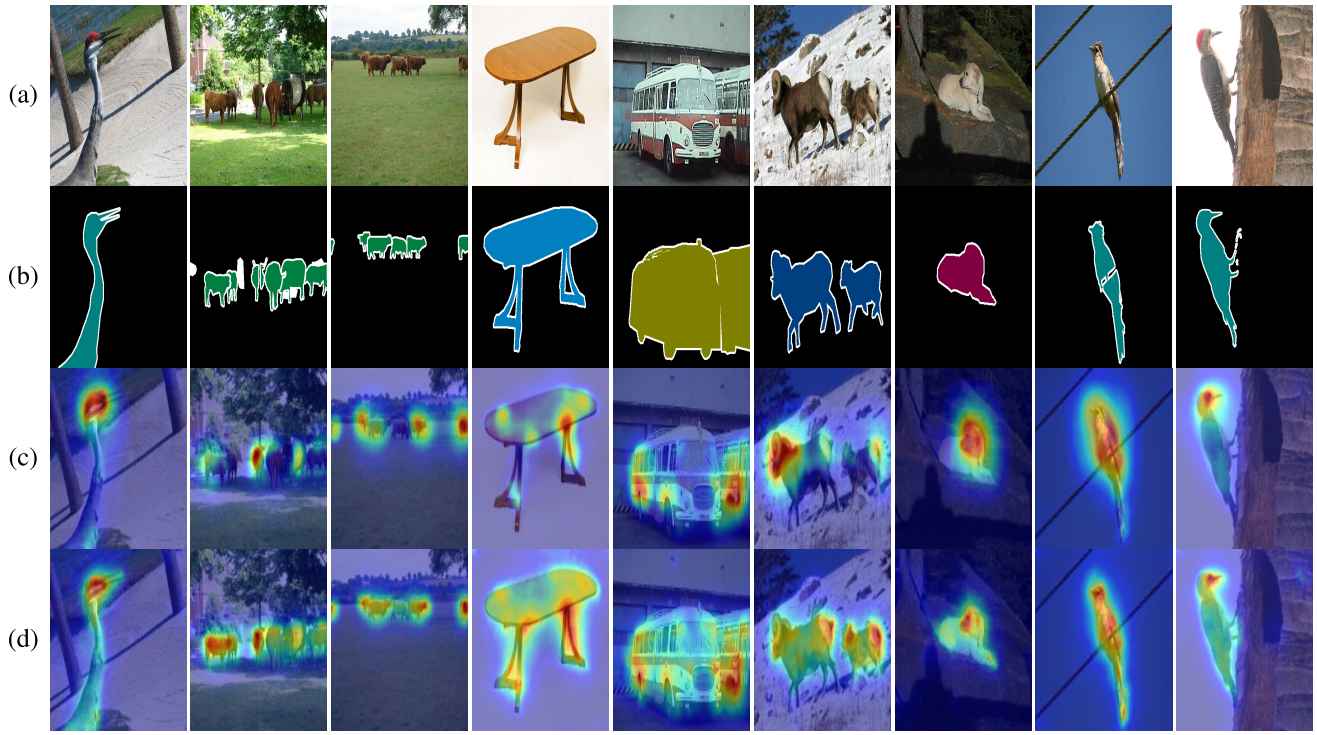


FIGURE 5. Qualitative results of the CAMs. (a) Image. (b) Ground truth. (c) The CAM of the previous method [16] with ResNet-50 as backbone. (d) The CAM of ours with ResNet-50 as backbone.

TABLE 1. Comparison of Fully-CAM and existing methods on the PASCAL VOC 2012 val datasets.

Method	bkg	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbk	person	plant	sheep	sofa	train	tv	mean
EM-Adapt [9]	67.2	29.2	17.6	28.6	22.2	29.6	47.0	44.0	44.2	14.6	35.1	24.9	41.0	34.8	41.6	32.1	24.8	37.4	24.0	38.1	31.6	33.8
CCNN [30]	68.5	25.5	18.0	25.4	20.2	36.3	46.8	47.1	48.0	15.8	37.9	21.0	44.5	34.5	46.2	40.7	30.4	36.3	22.2	38.8	36.9	35.3
MIL+seg [31]	79.6	50.2	21.6	40.9	34.9	40.5	45.9	51.5	60.6	12.6	51.2	11.6	56.8	52.9	44.8	42.7	31.2	55.4	21.5	38.8	36.9	42.0
SEC [32]	82.4	62.9	26.4	61.6	27.6	38.1	66.6	62.7	75.2	22.1	53.5	28.3	65.8	57.8	62.3	52.5	32.5	62.6	32.1	45.4	45.3	50.7
AdvErasing [22]	83.4	71.1	30.5	72.9	41.6	55.9	63.1	60.2	74.0	18.0	66.5	32.4	71.7	56.3	64.8	52.4	37.4	69.1	31.4	58.9	43.9	55.0
Affinity [13]	88.2	68.2	30.6	81.1	49.6	61.0	77.8	66.1	75.1	29.0	66.0	40.2	80.4	62.0	70.4	73.7	42.5	70.7	42.6	**68.1**	51.6	61.7
RRM [28]	87.9	75.9	31.7	78.3	54.6	62.2	80.5	73.7	71.2	30.5	67.4	40.9	71.8	66.2	70.3	72.6	49.0	70.7	38.4	62.7	**58.4**	62.6
SEAM [21]	88.8	68.5	**33.3**	**85.7**	40.4	67.3	78.9	76.3	81.9	29.1	75.5	**48.1**	79.9	73.8	71.4	**75.2**	48.9	79.8	40.9	58.2	53.0	64.5
SSDD [33]	89.0	62.5	28.9	83.7	52.9	59.5	77.6	73.7	87.0	**34.0**	**83.7**	47.6	**84.1**	77.0	73.9	69.6	29.8	**84.0**	**43.2**	68.0	53.4	64.9
BES [34]	88.9	74.1	29.8	81.3	53.3	69.9	**89.4**	79.8	84.2	27.9	76.9	46.6	78.8	75.9	72.2	70.4	50.8	79.4	39.9	65.3	44.8	65.7
Our(ResNet-50)	**89.9**	76.9	32.7	82.9	**62.8**	66.0	89.0	79.9	86.0	29.0	78.9	48.0	79.5	79.6	75.9	72.3	48.2	78.4	40.0	61.4	57.8	67.4
Our(ResNet-101)	89.9	**78.7**	31.4	85.0	54.9	**72.0**	89.3	**80.2**	**87.5**	29.8	80.4	47.8	83.0	**81.3**	**76.4**	71.3	**52.1**	81.1	42.7	64.5	52.6	**68.2**

To further study the advantages of plug and play of Fully-CAM, we mount our framework on multiple trained ordinary classifiers (see Table.5). It is easy to see that Fully-CAM can significantly improve the CAM of multiple backbones. Specially, there is a 2% improvement on VGG-16, and an increase of about 6% for ResNet-50 and ResNet-101 with our framework. Given this phenomenon, we speculate that this is related to the gradient. ResNet has the residual block, which can significantly alleviate the vanishing gradient problem. Although the VGG-16 network is not very deep, it may also be affected.

Based on the above ablation studies, Fully-CAM exploits the potential of the ordinary classifier and yields the best performance in CAM. Fig.5 illustrates the qualitative results between Fully-CAM and the traditional CAM based on ResNet-50. It can be seen that our method has a more complete and accurate localizing effect.

TABLE 2. The ablation study for the fusion of different convolution layers. Performance on PASCAL VOC 2012 train set. Stage 5: layer4.2.conv3 of ResNet-50. Stage 4: layer3.5.conv3 of ResNet-50. Stage 3: layer2.3.conv3 of ResNet-50. Other Convs: Remaining convolution layer.

Stage 5	Stage 4	Stage 3	Other Convs	CAM (%)
√	-	-	-	44.76
√	√	-	-	49.15
√	√	√	-	51.52
√	√	√	√	53.88

C. COMPARISON WITH EXISTING METHOD

Table.1 and Table.6 show the experimental results of our method and Existing methods. To improve the accuracy of pixel-level pseudo labels, we follow the previous works [12] to train an IRNet based on our revised CAM. The pseudo labels after IRNet and applying the dense Conditional

FIGURE 6. Qualitative results of the segmentation networks trained with pseudo pixel-level labels. Note that those pseudo labels are generated using only image-level labels. (a) Image. (b): Ground truth. (c): Segmentation results of baseline [12] with ResNet-50 as backbone. (d): Segmentation results of ours with ResNet-50 as backbone.

TABLE 3. Evaluation of various weakly supervised localization methods with semantic segmentation metric (mIoU).

Method	mIoU(%)
traditional CAM [16]	47.43
Grad-CAM [24]	46.53
Grad-CAM++ [35]	47.37
Our	53.88

TABLE 4. The ablation study for transformations in the inference on PASCAL VOC 2012 dataset with [0.5, 1, 1.5, 2.0]: different scale rates and flip: horizontal flipping.

1	0.5	1.5	2.0	flip	CAM (%)
√	-	-	-	-	51.28
√	√	-	-	-	51.96
√	√	√	-	-	52.99
√	√	√	√	-	53.83
√	√	√	√	√	53.88

TABLE 5. The improvements of extra performance and flexibility of plug and play.

backbone	CAM(%)	Our(%)
VGG-16	48.9	50.99
ResNet-50	48.3	53.88
ResNet-101	48.2	54.26

Random Field (dCRF) are used to train the semantic segmentation network DeepLab [36] with ResNet-101 for WSSS. As shown in Table.6, we achieve mIoU of 68.2% and 68.9%

TABLE 6. Comparison of Fully-CAM and existing methods on the PASCAL VOC 2012 val and test datasets with image-level class label.

Method	Backbone	Pub.	Val	Test
AffinityNet [13]	ResNet38	CVPR18	61.7	63.7
IRNet [12]	ResNet50	CVPR19	63.5	64.8
SSDD [33]	ResNet38	ICCV19	64.9	65.5
ICD [15]	ResNet101	CVPR20	64.1	64.3
SEAM [21]	ResNet38	CVPR20	64.5	65.7
Sub-category [14]	ResNet101	CVPR20	66.1	65.9
RRM [28]	ResNet101	AAAI20	66.3	66.5
BES [34]	ResNet101	ECCV20	65.7	66.6
CPN[40]	ResNet101	ICCV21	67.8	68.5
AdvCAM [41]	ResNet101	CVPR21	68.1	68.8
Puzzle-CAM [19]	ResNeSt101	ICIP21	66.9	67.7
CDA [42]	ResNet38	ICCV21	66.1	66.8
WSGCN [37]	ResNet101	ICME21	**68.7**	**69.3**
CGNet [38]	ResNet38	ICCV21	68.4	68.2
ECS-Net [39]	ResNet38	ICCV21	66.6	67.6
Our + IRNet	ResNet-50	-	67.4	68.1
Our + IRNet	ResNet-101	-	68.2	68.9

on PASCAL VOC 2012 val and test sets. Specially, the original IRNet with ResNet-50 only reaches 64.8% on test set, but when IRNet is applied to the CAM obtained by our method, the mIoU can be increased by 3.3%. It further reveals that CAM does limit the performance of WSSS. Moreover, ResNet-50 has fewer parameters than ResNet-101, but our method based on ResNet-50 is far better than most of the existing ResNet-101 methods. Table.7 shows that our method with only image-level information outperforms the most with extra supervision on the ResNet-101 as the

TABLE 7. Comparison of Fully-CAM and existing methods on the PASCAL VOC 2012 val and test datasets with extra supervised information.

Method	Backbone	Pub.	Val	Test
MCOF [43]	ResNet101	CVPR18	60.3	61.2
SeeNet [18]	ResNet101	NIPS18	63.1	62.8
DSRG [17]	ResNet101	CVPR18	61.4	63.2
FickleNet [27]	ResNet101	CVPR19	64.9	65.3
CIAN [20]	ResNet101	AAAI20	64.3	65.4
OAA+ [44]	ResNet101	ICCV19	65.2	66.4
EME [45]	ResNet101	ECCV20	67.2	66.7
MCIS [46]	ResNet38	ECCV20	66.2	66.9
ICD [15]	ResNet101	CVPR20	67.8	68.0
Group-WSSS [47]	ResNet101	AAAI21	68.2	68.5
DRS	ResNet101	AAAI21	71.2	71.4
PPC [48]	ResNet101	CVPR22	**72.6**	**73.6**
Our + IRNet	ResNet-50	-	67.4	68.1
Our + IRNet	ResNet-101	-	**68.2**	**68.9**

backbone. We can see from these tables that our method achieves a better performance than the most methods in mIoU, and illustrate that we have fully explored the potential of CAM.

Furthermore, qualitative comparison of the segmentation networks trained with pseudo pixel-level labels is shown in Fig.6. IRNet and our method use the same backbone ResNet-50 and the same training method. Obviously, we can see that the CAM obtained by our method dramatically improves semantic segmentation performance. The original segmentation effect (as shown in (c) in Fig.6) is rough, and many pixel labels are missing. rough and lacks many pixel labels. However, our work can greatly make up for these deficiencies, and our effect (as shown in (d) in Fig.6) is more complete and refined

Admittedly, although we have surpassed most of the current advanced methods, there is still a gap between us and the state-of-the-art work. Nevertheless, our research is one of the few to improve CAM's performance compare with other works [37], [38], [39]. CAM has always been an indispensable part of WSSS. We have experimentally proved that ordinary classifiers can exceed their original performance without additional training throught our method.

V. CONCLUSION

In this work, we first profoundly rethink CAM. We find that the reason for the poor localization ability of CAM is not that the classifier can only highlight the most discriminative regions but the insufficient use of information. Then, to fully explore the potential of the classifier, we visualize the CAM of each convolution layer of the classifier and find that the classifier can highlight whole object regions. Next, we propose Fully-CAM, designed as a plug-in unit to take all feature maps to participate in the localizing task. Without complex training, the ultimate CAM highlights the whole area of the object. Finally, our CAM is used in the previous work, which significantly improves the performance of the previous method on the PASCAL VOC 2012 dataset. In the future, we will make efforts in weakly supervised object detection with only image-level label. The reason is that CAM is necessary for weakly-supervised tasks with image level labels. We believe that the good CAMs obtained by our method can improve the performance of weakly supervised object detection.

ACKNOWLEDGMENT

(Yuqi Song and Xiaojie Li contributed equally to this work.)

REFERENCES

[1] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," 2014, *arXiv:1412.7062*.
[2] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 801–818.
[3] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.
[4] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.
[5] D. Lin, J. Dai, J. Jia, K. He, and J. Sun, "ScribbleSup: Scribble-supervised convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3159–3167.
[6] P. Vernaza and M. Chandraker, "Learning random-walk label propagation for weakly-supervised semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7158–7166.
[7] J. Dai, K. He, and J. Sun, "BoxSup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1635–1643.
[8] A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele, "Simple does it: Weakly supervised instance and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 876–885.
[9] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille, "Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1742–1750.
[10] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei, "What's the point: Semantic segmentation with point supervision," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 549–565.
[11] K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool, "Deep extreme cut: From extreme points to object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 616–625.
[12] J. Ahn, S. Cho, and S. Kwak, "Weakly supervised learning of instance segmentation with inter-pixel relations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2209–2218.
[13] J. Ahn and S. Kwak, "Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4981–4990.
[14] Y.-T. Chang, Q. Wang, W.-C. Hung, R. Piramuthu, Y.-H. Tsai, and M.-H. Yang, "Weakly-supervised semantic segmentation via sub-category exploration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8991–9000.
[15] J. Fan, Z. Zhang, C. Song, and T. Tan, "Learning integral objects with intra-class discriminator for weakly-supervised semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4283–4292.
[16] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.
[17] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang, "Weakly-supervised semantic segmentation network with deep seeded region growing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7014–7023.
[18] Q. Hou, P. Jiang, Y. Wei, and M.-M. Cheng, "Self-erasing network for integral object attention," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–11.

[19] S. Jo and I.-J. Yu, "Puzzle-CAM: Improved localization via matching partial and full features," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2021, pp. 639–643.
[20] J. Fan, Z. Zhang, T. Tan, C. Song, and J. Xiao, "CIAN: Cross-image affinity net for weakly supervised semantic segmentation," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 10762–10769.
[21] Y. Wang, J. Zhang, M. Kan, S. Shan, and X. Chen, "Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12275–12284.
[22] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan, "Object region mining with adversarial erasing: A simple classification to semantic segmentation approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1568–1576.
[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[24] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.
[25] S. Yang, Y. Kim, Y. Kim, and C. Kim, "Combinational class activation maps for weakly supervised object localization," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 2941–2949.
[26] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.
[27] J. Lee, E. Kim, S. Lee, J. Lee, and S. Yoon, "FickleNet: Weakly and semi-supervised semantic image segmentation using stochastic inference," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5267–5276.
[28] B. Zhang, J. Xiao, Y. Wei, M. Sun, and K. Huang, "Reliability does matter: An end-to-end weakly supervised semantic segmentation approach," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 12765–12772.
[29] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 991–998.
[30] D. Pathak, P. Krahenbuhl, and T. Darrell, "Constrained convolutional neural networks for weakly supervised segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1796–1804.
[31] P. O. Pinheiro and R. Collobert, "From image-level to pixel-level labeling with convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1713–1721.
[32] A. Kolesnikov and C. H. Lampert, "Seed, expand and constrain: Three principles for weakly-supervised image segmentation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 695–711.
[33] W. Shimoda and K. Yanai, "Self-supervised difference detection for weakly-supervised semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5208–5217.
[34] L. Chen, W. Wu, C. Fu, X. Han, and Y. Zhang, "Weakly supervised semantic segmentation with boundary exploration," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 347–362.
[35] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 839–847.
[36] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2017.
[37] S.-Y. Pan, C.-Y. Lu, S.-P. Lee, and W.-H. Peng, "Weakly-supervised image semantic segmentation using graph convolutional networks," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2021, pp. 1–6.
[38] H. Kweon, S.-H. Yoon, H. Kim, D. Park, and K.-J. Yoon, "Unlocking the potential of ordinary classifier: Class-specific adversarial erasing framework for weakly supervised semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6994–7003.
[39] K. Sun, H. Shi, Z. Zhang, and Y. Huang, "ECS-Net: Improving weakly supervised semantic segmentation by using connections between class activation maps," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 7283–7292.
[40] F. Zhang, C. Gu, C. Zhang, and Y. Dai, "Complementary patch for weakly supervised semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 7242–7251.
[41] J. Lee, E. Kim, and S. Yoon, "Anti-adversarially manipulated attributions for weakly and semi-supervised semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4071–4080.
[42] Y. Su, R. Sun, G. Lin, and Q. Wu, "Context decoupling augmentation for weakly supervised semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 7004–7014.
[43] X. Wang, S. You, X. Li, and H. Ma, "Weakly-supervised semantic segmentation by iteratively mining common object features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1354–1362.
[44] P.-T. Jiang, Q. Hou, Y. Cao, M.-M. Cheng, Y. Wei, and H. Xiong, "Integral object mining via online attention accumulation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2070–2079.
[45] J. Fan, Z. Zhang, and T. Tan, "Employing multi-estimations for weakly-supervised semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 332–348.
[46] G. Sun, W. Wang, J. Dai, and L. Van Gool, "Mining cross-image semantics for weakly supervised semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 347–365.
[47] X. Li, T. Zhou, J. Li, Y. Zhou, and Z. Zhang, "Group-wise semantic mining for weakly supervised semantic segmentation," 2020, *arXiv:2012.05007*.
[48] Y. Du, Z. Fu, Q. Liu, and Y. Wang, "Weakly supervised semantic segmentation by pixel-to-prototype contrast," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4320–4329.

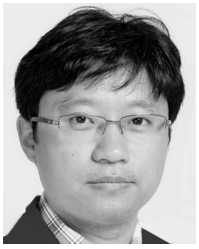
YUQI SONG is currently pursuing the M.S. degree in computer science and technology with the Chengdu University of Information Technology. His current research interests include weakly-supervised semantic segmentation and semantic segmentation of medical images.

XIAOJIE LI was born in 1981. She received the Ph.D. degree in computer science and engineering from the College of Computer Science, Sichuan University, Chengdu, China. She is currently an Associate Professor (a member of CCF) with the College of Computer Science, Chengdu University of Information Technology, Chengdu. Her research interests include machine learning, neural networks, and data mining.

CANGHONG SHI was born in 1984. He received the B.S. degree in mathematics and applied mathematics from Hebei Normal University, Shijiazhuang, China, in 2009, and the M.S. degree in basic mathematics from the Chengdu University of Information Technology, Chengdu, China, in 2014. He is currently pursuing the Ph.D. degree with Southwest Jiaotong University, Chengdu. He is currently working with Xihua University. His current research interests include multimedia information security and digital audio signal forensics.

SHIHAO FENG was born in 1998. He received the Bachelor of Science degree in computer science from The University of Auckland, Chongqing, China. He is currently pursuing the master's degree in data science with The University of Auckland, Auckland, New Zealand.

XIN WANG (Senior Member, IEEE) received the Ph.D. degree in computer science from the University at Albany, State University of New York, in 2015. He is currently a Research Affiliate at the University at Buffalo, State University of New York. His research interests include machine learning, reinforcement learning, deep learning, and their applications.



YONG LUO received the Ph.D. degree from Sichuan University, Chengdu, China. He is a member of the Radiotherapy Group of Glioma Special Committee of Chinese Medical Association, a Young Member of the Radiotherapy Professional Committee of Wu Jieping Medical Foundation, a Supervisor of the Sichuan Cancer Society, a Standing Committee Member of the Glioma Special Committee of Sichuan Cancer Society, a member of the NPC Special Committee of Sichuan Cancer Society, and a member of the Tumor Rehabilitation Committee of Sichuan Geriatrics Association. He is currently working with the West China Hospital of Sichuan University.

XI WU received the Ph.D. degree from Southwest Jiaotong University. He is currently a Professor and the Deputy Dean of the Department of Computer Science, Chengdu University of Information Technology. He is also the Deputy Director of the Collaborative Innovation Center for Image and Geospatial Information of Sichuan Province, China. His main research interest includes computational intelligence cooperated with cognitive studies. He is also interested in the area of novel methods for analysis of imaging data after joined the Department of Computer Science, Chengdu University of Information Technology.