Complementary Patch for Weakly Supervised Semantic Segmentation

Fei Zhang, Chaochen Gu[*] Chenyue Zhang
Shanghai Jiao Tong University, China
{ferenas, jacygu, lucklypeach}@sjtu.edu.cn

Yuchao Dai
Northwestern Polytechnical University, China
daiyuchao@nwpu.edu.cn

Abstract

Weakly Supervised Semantic Segmentation (WSSS) based on image-level labels has been greatly advanced by exploiting the outputs of Class Activation Map (CAM) to generate the pseudo labels for semantic segmentation. However, CAM merely discovers seeds from a small number of regions, which may be insufficient to serve as pseudo masks for semantic segmentation. In this paper, we formulate the expansion of object regions in CAM as an increase in information. From the perspective of information theory, we propose a novel Complementary Patch (CP) Representation and prove that the information of the sum of the CAMs by a pair of input images with complementary hidden (patched) parts, namely CP Pair, is greater than or equal to the information of the baseline CAM. Therefore, a CAM with more information related to object seeds can be obtained by narrowing down the gap between the sum of CAMs generated by the CP Pair and the original CAM. We propose a CP Network (CPN) implemented by a triplet network and three regularization functions. To further improve the quality of the CAMs, we propose a Pixel-Region Correlation Module (PRCM) to augment the contextual information by using object-region relations between the feature maps and the CAMs. Experimental results on the PASCAL VOC 2012 datasets show that our proposed method achieves a new state-of-the-art in WSSS, validating the effectiveness of our CP Representation and CPN.

1. Introduction

Thanks to the booming of deep learning methods, recent years have witnessed extraordinary progress in semantic segmentation [28, 43, 7, 8]. However, the prerequisite of a successful neural network for semantic segmentation is pixel-level segmentation ground-truth, which requires massive investments in manual annotation. Numerous efforts have been devoted to developing Weakly Supervised Semantic Segmentation (WSSS) to ease the pressure, which aims to train a semantic segmentation network by using weaker supervision, such as image-level classification labels [22, 5, 2, 3], bounding boxes [21, 10], scribbles [27] and points [4]. Image-level labels, as the most conveniently-acquired annotation format, have been extensively studied in WSSS. In this work, we particularly focus on WSSS using image-level labels.

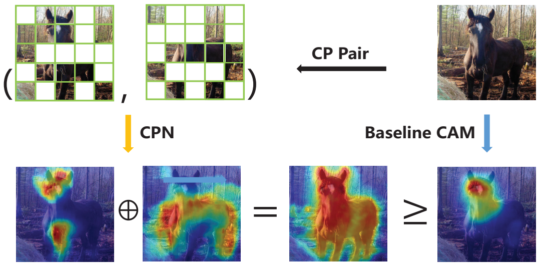

Figure 1. Illustration of our proposed method. The original CAM simply finds object seeds in most discriminative regions. To enlarge the seed areas, our Complementary Patch Network (CPN) uses a pair of images with CP regions (CP Pair) to generate two CAMs, the sum of which are supposed to incorporate more information of the foreground than the original CAM.

Most WSSS approaches generating initial seeds through image-level labels heavily rely on an efficient method—Class Activation Map (CAM) [45]. Nevertheless, this architecture appears to be barely sensitive to the most discriminative regions, resulting in many incomplete foreground areas. To address the issue, a promising way is to erase or ignore some high response regions to help CAM 'see' more seeds in an image [36, 24, 23, 9], *i.e.*, region erasing or mining methods. However, these methods are more or less losing part of the regions of an image in each training epoch due to the randomness of the hiding process. It seems to be effective to intentionally cover the high response areas identified from the CAM in each training epoch, while such iterative operation introduces much computational complexity, and it is difficult to properly determine the number of iterations for each image as well.

In this paper, we show that CAM could explore more high response areas by taking full advantage of the information of an image, specifically including both uncov-

[*] Corresponding author

ered and hidden parts. Based on the motivation, we treat the task of expanding object seeds in CAM as an increase in information, and develop a simple yet efficient concept—Complementary Patch (CP) Representation: the self-information of the CAM of an image is less than or equal to the sum of the self-information of the CAMs, which are obtained by CP Pair, namely two images with CP regions. Therefore, an improved CAM could be obtained by adding up the CAMs generated by the CP Pair (shown in Fig. 1). In addition, we show that the equality holds under two extreme cases. One is that if the patch size is too large, one of the CP Pair equals the original image, and the other is that two images in the CP Pair are almost the same for the network if the patch size is too small. Under these extreme conditions, the CP Pair is unable to seek out new seed areas compared to the original image. Thus the degree of the increase in information (object seeds) is subject to the patch size for the CP Pair.

Building upon the CP Representation, we propose a CP Network (CPN) to narrow down the gap between the improved CAM mentioned above and the one by the original image. CPN is formed by a triplet network with Triplet CP (TCP) loss and CP Cross Regularization (CPCR) loss, serving as minimizing the above discrepancy. For the generation of the CP Pair, we propose to use grid (Grid Patch) or superpixel (Super-pixel Patch) as the patch template. Furthermore, CPN introduces a Pixel-Region Correlation Module (PRCM), which aims to capture the relationship between the pixels and regions, and incorporates it with Pixel Correlation Module (PCM) [35] to further improve the consistency of the predicted CAM.

Extensive experiments conducted on PASCAL VOC 2012 [13] demonstrate the effectiveness of our CPN. As a result, our model yields a new state-of-the-art performance by 67.8% and 68.5% on the *val* set and *test* set. Furthermore, we notice that the performance of our CPN is influenced by the patch size, which is in accordance with our analysis of the CP Representation in extreme cases.

Our main contributions are summarized as three-fold:

- We propose a simple yet effective Complementary Patch (CP) representation to enlarge the seed regions in CAM, which narrows down the gap between the original CAM and the CAMs by summing up the CAMs of the CP pair.

- Building upon the CP representation, we present a triplet network (CPN) with Triplet CP (TCP) loss and CP Cross Regularization (CPCR). Moreover, a Pixel-Region Correlation Module (PRCM) is proposed to further refine the CAM.

- Experimental results on the PASCAL VOC 2012 show that our proposed framework achieves state-of-the-art performance in WSSS.

2. Related Work

Weakly supervised semantic segmentation As the most economic form in WSSS, image-level supervision has gained increasing attention from academia and industry. Recent advanced methods focus on modifying the seed areas produced by Class Activation Map (CAM) [45]. The first category [12, 22, 30] is dedicated to pooling-based methods to overcome the drawbacks led by Global Max-Pooling (GMP) and Global Average-Pooling (GAP), which are used to aggregate score maps into a classification score. SPN [30] proposes to regard super-pixel segmentation of the input image as the pooling module. The second category [20, 2, 35, 3, 31, 1, 38] investigates the inter-pixel or semantic relationship to expand the seed areas or remove the wrong seeds [42]. AffinityNet [2] proposes to learn the similarity between pixels and applies Random Walk (RW) to further refine the seed areas. The third category concentrates on making efficient use of extra easily-obtained resources, including web images [18],videos [18, 25] and saliency maps [37]. The fourth category turns to region erasing or mining methods, aiming to mark out more object regions in CAM by erasing or mining some high response regions. Adversarial Erasing [36] aims to explore more object seeds by iteratively erasing the discriminative regions detected by the original CAM from the image. However, it is hard to decide the exact number of iterations for each image. Attention-based Dropout Layer [9] is a tool that highlights the potential points by thresholding the attention maps obtained from the feature maps. To expand the seed areas, FickleNet [24] calculates the final score maps by randomly selecting the hidden units in the feature maps. As a data augmentation, Hide-and-Seek (HAS) [23] enlarges the seed areas by randomly hiding grid patches in each image. Nevertheless, these hiding methods above are incapable of using the entire information in an image during each training epoch. To excavate full information in an image as much as possible, we propose Complementary Patch (CP) Representation and design the CPN to support CAM mine out more foreground seeds.

Self-attention model For the sake of improving the quality of segmentation masks, models based on self-attention [32], refining the feature maps by use of context feature, is widely employed in various segmentation networks. Wang *et al*. [33] proposes non-local block to produce an attention map by taking account of the correlation between each spatial point in the feature maps. To further enrich the contextual information, DANet [17] combines two self-attention modules, namely channel attention and spatial attention. Yuan *et al*. [40] proposes the object-contextual representations to identify a pixel by using its corresponding object class, reinforcing the object contextual information.

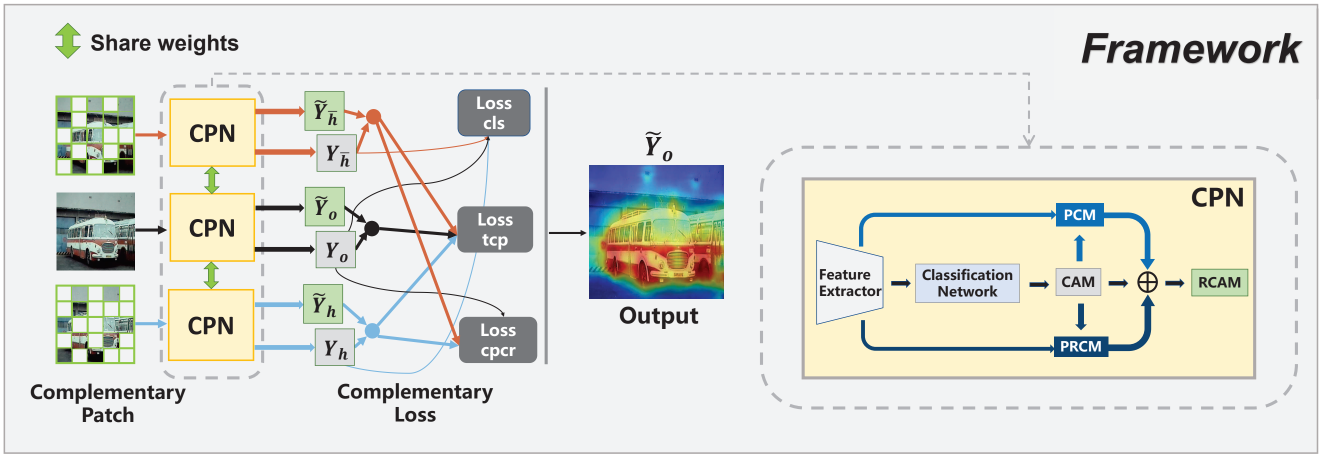

Figure 2. The overall framework of our method. The whole structure of CPN is a triplet network with three branches, jointly feeding the original image (the black flow) and the CP Pair (the red and blue flows). PCM and the proposed PRCM collectively improve the quality of the original CAM to the refined CAM (RCAM). Finally, all outputs are constrained with three losses, which are $\mathcal{L}_{cls}$, $\mathcal{L}_{tcp}$ and $\mathcal{L}_{cpcr}$. The dot (the red, blue, or black one) means that both outputs connected to it are leveraged in the following loss. During inference, the RCAM from the original image ($\widetilde{\boldsymbol{Y}}_o$) is used to predict the mask for segmentation.

3. Proposed Methods

3.1. Complementary Patch Representation

Let us denote the CAM of an image $\boldsymbol{I}$ of size $3 \times H \times W$ as $\boldsymbol{Y} \in \mathbb{R}^{C \times H \times W}$, where C refers to the number of objects (including background). The generation of $\boldsymbol{Y}$ typically begins with training a multi-label classification network, comprising a feature extractor layer, a Global Average-Pooling (GAP) layer, and a classification layer. Thus $\boldsymbol{Y}$ related to the c-th object, denoted as $\boldsymbol{Y}^c$, can be gained by:

$$\boldsymbol{Y}^c = (\boldsymbol{\theta}^c)^\top \boldsymbol{f}, \tag{1}$$

where $\boldsymbol{f} \in \mathbb{R}^{C_f \times H \times W}$ with C_f channels is the feature maps from the final layer, and $\boldsymbol{\theta}^c \in \mathbb{R}^{C_f \times 1}$ is the corresponding classifier weight of c-th class in the classification layer. Denote $\boldsymbol{I}$ with some hidden units as $\boldsymbol{I}_h \in \mathbb{R}^{3 \times H \times W}$ and its counterpart with complementary hidden regions as $\boldsymbol{I}_{\overline{h}} \in \mathbb{R}^{3 \times H \times W}$. Intuitively, we have $\boldsymbol{I}_{\boldsymbol{h}} + \boldsymbol{I}_{\overline{\boldsymbol{h}}} = \boldsymbol{I}$. For convenience, we name $(\boldsymbol{I}_h, \boldsymbol{I}_{\overline{h}})$ as the Complementary Patch Pair (CP Pair) of $\boldsymbol{I}$.

According to the previous experiments of the region and erasing methods [23, 36], both $\boldsymbol{I}_h$ and $\boldsymbol{I}_{\overline{h}}$ may individually help CAM dig out more potential areas. However, some parts in $\boldsymbol{I}$ are apparently ignored if simply using $\boldsymbol{I}_h$ or $\boldsymbol{I}_{\overline{h}}$ in each training epoch since each image could only be used once during one epoch. To make full use of the information in $\boldsymbol{I}$, here we propose to use the CP Pair to find more seeds in $\boldsymbol{Y}^c$. For a partitioned image in Fig. 3, suppose $\boldsymbol{I}$ is split into N patches and there are two kinds of patch regions for the c-th class, $\Omega = \{\mathcal{A}_i^c\}_{i=1}^{N_a}$ and $\Gamma = \{\mathcal{D}_j^c\}_{j=1}^{N_d}$, where $N_a + N_d = N$. $\mathcal{A}^c$ represents the patch region that contains the seeds of the object c, while $\mathcal{D}^c$ covers no seeds related to it. We denote $p_c(x), x \in \{\Omega \cup \Gamma\}$ as the probability function of finding c-th seeds in x. Higher $p_c(x)$ refers to more seeds in patch x of $\boldsymbol{I}$. Therefore, we have $\sum_{i=1}^{N_a} p_c(x = \mathcal{A}_i^c) = 1$ and $p_c(x = \mathcal{D}_j^c) = 0, j \in \{1, 2, ..., N_d\}$. Under the definition above, the self information of $\boldsymbol{Y}^c$ is denoted as $\mathcal{H}(\boldsymbol{Y}^c)$ and expressed as:

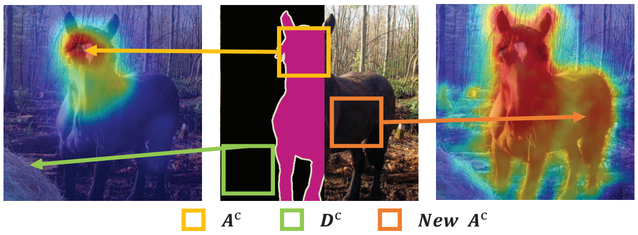

Figure 3. Two kinds of patch regions in an image. $\mathcal{A}^c$ refers to the region that covers the seeds related to object c, and $\mathcal{D}^c$ contains no seeds of it. The baseline CAM (left) can merely recognize part of $\mathcal{A}^c$ regions, while the new CAM (right) by the CP Pair can find ***New*** $\mathcal{A}^c$.

$$\mathcal{H}(\boldsymbol{Y}^c) = -\sum_{x \in \Omega_y} \log(p_c(x)), \tag{2}$$

where $\Omega_y \subset \Omega$ refers to the set of $\mathcal{A}^c$ in $\boldsymbol{Y}^c$. Note that the ground truth of the c-th class contains all patch $\mathcal{A}^c$, so it is said to have the maximum information. Our aim is to increase $\mathcal{H}(\boldsymbol{Y}^c)$ by increasing $|\Omega_y|$.

Meanwhile, let $\mathcal{H}(\boldsymbol{Y}_h^c)$ and $\mathcal{H}(\boldsymbol{Y}_{\overline{h}}^c)$ be the information of the CAMs generated by the CP Pair, which can be jointly calculated as follows:

$$\mathcal{H}(\boldsymbol{Y}_q^c) = -\sum_{x \in \Omega_q} \log(p_c(x)),\ q \in \{h, \overline{h}\}, \tag{3}$$

where Ω_h ($\Omega_{\overline{h}}$) $\subset \Omega$ is the set of $\mathcal{A}^c$ in $\boldsymbol{Y}^c_h$ ($\boldsymbol{Y}^c_{\overline{h}}$). Randomly covering parts of the object c in $\boldsymbol{I}$ leads to an increase of $\boldsymbol{New}\,\mathcal{A}^c$. Unfortunately, it is undecidable to straightly compare $|\Omega_y|$ with $|\Omega_h|$ or $|\Omega_{\overline{h}}|$, because some discriminative parts in Ω_y are possibly hidden in $\boldsymbol{Y}^c_h$ or $\boldsymbol{Y}^c_{\overline{h}}$. Due to the complementary attribution ($\Omega_h \cap \Omega_{\overline{h}} = \varnothing$), however, the sum of $\mathcal{A}^c$ in $\boldsymbol{Y}^c_h$ and $\boldsymbol{Y}^c_{\overline{h}}$ contains the original high response regions from the baseline CAM, and the $\boldsymbol{New}\,\mathcal{A}_c$ regions sought by the CP Pair. Therefore, we have:

$$\Omega_h \cup \Omega_{\overline{h}} = \Omega_y \cup \Omega', \tag{4}$$

where $\Omega' \subset \Omega$ refers to the set of $\boldsymbol{New}\,\mathcal{A}^c$ and $\Omega' \cap \Omega_y = \varnothing$. Note that $\Omega' = \varnothing$ if one of the following extreme conditions holds:

1) $\boldsymbol{I} = \boldsymbol{I}_h$ or $\boldsymbol{I} = \boldsymbol{I}_{\overline{h}}$. One of the CP Pair is equal to the original image when the patch size equals the image size;

2) $\boldsymbol{Y}^c_h \approx \boldsymbol{Y}^c_{\overline{h}}$. It is difficult for the classification net to discriminate the CP Pair if the patch size is too small, resulting in $\Omega_h = \Omega_{\overline{h}} = \Omega_y$.

Based on (4), we have that:

$$\begin{aligned} &\mathcal{H}(\boldsymbol{Y}^c_h) + \mathcal{H}(\boldsymbol{Y}^c_{\overline{h}}) \\ &= -\sum_{x\in\Omega_h} \log(p_c(x)) - \sum_{x\in\Omega_{\overline{h}}} \log(p_c(x)) \\ &= -\sum_{x\in\Omega_y} \log(p_c(x)) - \sum_{x\in\Omega'} \log(p_c(x)) \geq \mathcal{H}(\boldsymbol{Y}^c). \end{aligned} \tag{5}$$

According to (5), it is concluded that, except for two extreme cases, the sum of CAMs by the CP Pair is able to find more foreground seeds than $\boldsymbol{Y}^c$. To achieve an improved CAM, we propose the CP regularization with a pair of chosen parameters $\lambda \in [0,1], \overline{\lambda} = 1 - \lambda$ as follows:

$$||(\lambda \boldsymbol{Y}^c_h + \overline{\lambda}\boldsymbol{Y}^c_{\overline{h}}) - \boldsymbol{Y}^c||_1. \tag{6}$$

Denote the number of patches hidden in $\boldsymbol{I}_h$ as N_h. Then λ can be obtained by $\lambda = 1 - N_h/N$, meaning that the weight is decided by the quantity of uncovered pixels in $\boldsymbol{I}_h$. To corporate (6) on the original classification net, we turn to a shared-weights triplet network as shown in Fig. 2, namely CP Network (CPN). One branch deals with input $\boldsymbol{I}$ and outputs $\boldsymbol{Y}^c$, while the other two branches respectively generate ($\boldsymbol{Y}^c_h, \boldsymbol{Y}^c_{\overline{h}}$) made by the CP Pair. Note that we stop the gradient update for ($\boldsymbol{Y}^c_h, \boldsymbol{Y}^c_{\overline{h}}$) to push $\boldsymbol{Y}^c$ to approximate the better one. In this way, these three outputs are supposed to be regularized by (6).

3.2. Complementary Patch Strategies

Grid Patch [23, 44] is a common method that can be applied to generate a bunch of $\boldsymbol{I}_h$ for $\boldsymbol{I}$. Specifically, a grid patch with a fixed size of $S \times S \times 3$ can partition $\boldsymbol{I}$ into $H \times W/(S \times S)$ patches. Then each patch hidden with a probability $p_h = 0.5$ is fed into the classification net. Following [23], S is evenly chosen from a set K of fixed numbers to fit the size of different objects. To guarantee the same data distribution between the training and the testing sets, the value of the hidden pixels is set to equal the mean RGB values of the images among the whole training sets.

On the other hand, the super-pixel region contains rich information about an image. Therefore, we also propose a Super-pixel Patch strategy, which uses the super-pixels generated by SLIC [16] as the patches. Here the number of the super-pixels depends on a predefined segment number, denoted as S_N. Note that we test these two patch strategies respectively in experiments.

3.3. Modules in CPN

We propose the CP Representation to help CAM find more foreground seeds. However, the regularization in (6) is incapable of sufficiently improving the CAM merely by using the output of the typical network. To further refine the original CAMs, [35] proposes a modified self-attention module named Pixel Correlation Module (PCM), capturing contextual information by taking advantage of pixel relationships in feature maps. Here we take a brief introduction to the self-attention module [33], which can be normally expressed as:

$$\boldsymbol{Y}_{out} = \frac{\mu(\boldsymbol{X_{in}})\mathcal{J}(\boldsymbol{X_{in}})}{\sum_{i=1}^{HW}\sum_{j=1}^{HW}\mathcal{J}(\boldsymbol{X_{in}})_{ij}} + \boldsymbol{X_{in}}, \tag{7}$$

$$\mathcal{J}(\boldsymbol{X_{in}}) = e^{g(\boldsymbol{X_{in}})^{\mathsf{T}}\delta(\boldsymbol{X_{in}})}, \tag{8}$$

where $\boldsymbol{X}_{in}$ and $\boldsymbol{Y}_{out}$ are respectively the input and output feature. $\mathcal{J}$ is used for measuring the relationship between the adjacent pixels, and μ provides a representation of each pixel in $\boldsymbol{X}_{in}$. Specially, μ, δ and g are implemented by is implemented by a 1×1 convolution layer.

Based on (7) and (8), the PCM refines the CAM $\boldsymbol{Y} \in \mathbb{R}^{C\times HW}$ (flattened into matrix formats) as:

$$\boldsymbol{Y}_{pcm} = \frac{\boldsymbol{Y}\mathcal{J}(\boldsymbol{X})}{\sum_{i=1}^{HW}\sum_{j=1}^{HW}\mathcal{J}(\boldsymbol{X})_{ij}}, \tag{9}$$

$$\mathcal{J}(\boldsymbol{X}) = \mathrm{ReLU}(\frac{g(\boldsymbol{X})^{\mathsf{T}}g(\boldsymbol{X})}{||g(\boldsymbol{X})||_1{}^2}), \tag{10}$$

where $\boldsymbol{X} \in \mathbb{R}^{C1\times HW}$ is the aggregation of some features in the classification net, and $\mathcal{J} : \mathbb{R}^{C_1} \mapsto \mathbb{R}^{HW}$ refers to the cosine distance to measure the inter-pixel feature similarity. Then we can obtain a refined CAM, denoted as $\boldsymbol{Y}_{pcm} \in \mathbb{R}^{C\times H\times W}$ (reshaped from $\boldsymbol{Y}_{pcm} \in \mathbb{R}^{C\times HW}$).

Object Contextual Representation (OCR) [40] is an effective approach to augment the contextual information based on exploring object-pixel relation. Therefore, we propose a Pixel-Region Correlation Module (PRCM) to help further improve the CAMs. Firstly, an Object-Region

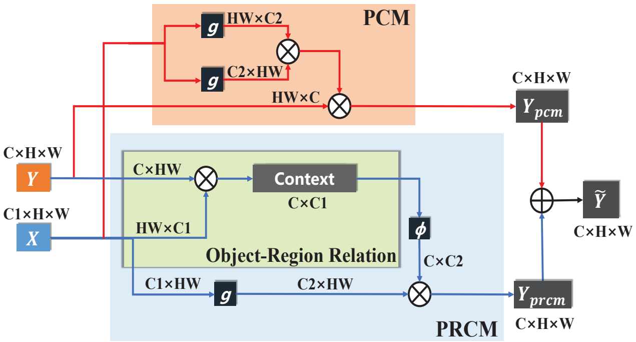

Figure 4. The structures of PCM (the red stream) and proposed PRCM (the blue stream). The final refined CAM $\widetilde{\boldsymbol{Y}}$ is the sum of $\boldsymbol{Y}_{pcm}$ and $\boldsymbol{Y}_{prcm}$.

Relation matrix $\boldsymbol{Z} \in \mathbb{R}^{C\times C_1}$ is represented by $\boldsymbol{Z} = \mathrm{SoftMax}(\boldsymbol{Y})\boldsymbol{X}^{\mathsf{T}}$. Here we directly treat $\boldsymbol{Y}$ as soft object regions, which are supposed to be the coarse segmentation maps corresponding to C objects [40]. Then we can obtain a Pixel-Region Relation $\boldsymbol{P_R} \in \mathbb{R}^{C\times H\times W}$ (reshaped from $\boldsymbol{P_R} \in \mathbb{R}^{C\times HW}$) as:

$$\boldsymbol{P_R} = \phi(\boldsymbol{Z})g(\boldsymbol{X}), \tag{11}$$

$$\boldsymbol{Y}_{prcm} = \boldsymbol{Y} \circ \mathrm{SoftMax}(\boldsymbol{P_R}), \tag{12}$$

where $\phi : \mathbb{R}^{C_1} \mapsto \mathbb{R}^{C_2}$ is also an embedding function similar to g. Since $\boldsymbol{P_R}$ represents the relation between regions in $\boldsymbol{X}$ and pixels in $\boldsymbol{Y}$ [40], we strengthen $\boldsymbol{Y}$ by (12) to gain the refined CAM, denoted as $\boldsymbol{Y}_{prcm} \in \mathbb{R}^{C\times H\times W}$.

In order to combine the contextual information collected by PCM and PRCM, the final smoothed CAM, denoted as $\widetilde{\boldsymbol{Y}} \in \mathbb{R}^{C\times H\times W}$, is the sum of $\boldsymbol{Y}_{prcm}$ and $\boldsymbol{Y}_{pcm}$. Fig. 4 illustrates the structures of PCM and PRCM. Thus, a pair of output CAMs $(\widetilde{\boldsymbol{Y}}, \boldsymbol{Y})$ can be readily generated for each branch in CPN. For convenience, we denote the CAMs in branch with $\boldsymbol{I}$ as $(\widetilde{\boldsymbol{Y}}_o, \boldsymbol{Y}_o)$, and the CAMs in branch with the CP Pair are respectively denoted as $(\widetilde{\boldsymbol{Y}}_h, \boldsymbol{Y}_h)$ and $(\widetilde{\boldsymbol{Y}}_{\overline{h}}, \boldsymbol{Y}_{\overline{h}})$.

3.4. Loss in CPN

Following conventions, an additional GAP layer is applied on the CAM, aggregating it into image-level prediction scores $s \in \mathbb{R}^{(C-1)\times 1}$. Note that s simply contains $C-1$ foreground objects since the image-level supervision lacks the background label. Therefore, we can obtain the score maps generated from $\boldsymbol{Y}_o$, $\boldsymbol{Y}_h$ and $\boldsymbol{Y}_{\overline{h}}$,which are respectively denoted as s_o,s_h and $s_{\overline{h}}$. Then we employ Multi-Label Soft Margin Loss $\boldsymbol{l}_{cls}$ for supervision:

$$\mathcal{L}_{cls} = \frac{1}{3}(\boldsymbol{l}_{cls}(s_o) + \boldsymbol{l}_{cls}(s_h) + \boldsymbol{l}_{cls}(s_{\overline{h}})). \tag{13}$$

Meanwhile, the CP Representation is adopted into the CPN for mining out more seeds, which is represented as:

$$\begin{aligned}\mathcal{L}_{tcp} =& ||(\lambda \boldsymbol{Y}_h + \overline{\lambda}\boldsymbol{Y}_{\overline{h}}) - \boldsymbol{Y}_o||_1 + \\ & ||(\lambda \widetilde{\boldsymbol{Y}}_h + \overline{\lambda}\widetilde{\boldsymbol{Y}}_{\overline{h}}) - \widetilde{\boldsymbol{Y}}_o||_1,\end{aligned} \tag{14}$$

Here the Triplet CP (TCP) loss, denoted as $\mathcal{L}_{tcp}$, is proposed based on the regularization in (6). Note that there are six output CAMs in the CPN since each branch posses two of them. Consequently, the TCP loss builds a connection among these six CAMs. Similarly with [35], to address the problem that $\widetilde{\boldsymbol{Y}}$ predicts all the pixels as the same class (mostly background), we introduce the CP Cross Regularization (CPCR) loss as:

$$\begin{aligned}\mathcal{L}_{cpcr} =& ||(\boldsymbol{Y}_o - \lambda \boldsymbol{Y}_h) - \overline{\lambda}\widetilde{\boldsymbol{Y}}_{\overline{h}}||_1 + \\ & ||(\boldsymbol{Y}_o - \overline{\lambda}\boldsymbol{Y}_{\overline{h}}) - \lambda\widetilde{\boldsymbol{Y}}_h||_1,\end{aligned} \tag{15}$$

Here we jointly regularize the refined CAMs by the CP Pair to make indirect effects on $\widetilde{\boldsymbol{Y}}_o$. As for the example of regularization of $\lambda\widetilde{\boldsymbol{Y}}_h$, it might be intuitive to regularize $\widetilde{\boldsymbol{Y}}_h$ by $\boldsymbol{Y}_h$. However, such direct regularization leads to a further degradation in our early experiments. Therefore, we use the gap between $\boldsymbol{Y}_o$ and $\lambda\boldsymbol{Y}_{\overline{h}}$ to regularize $\widetilde{\boldsymbol{Y}}_h$ in light of (6).

During the training, the background activation map is assessed by:

$$\boldsymbol{Y}^{c=0}(x,y) = (1 - \max_{1\le c\le C-1} \boldsymbol{Y}^c(x,y))^{\alpha}, \tag{16}$$

Here $\boldsymbol{Y}^c(x,y)$ is the activation value of category c at the position (x,y) in $\boldsymbol{Y} \in \{\boldsymbol{Y}_o, \boldsymbol{Y}_h, \boldsymbol{Y}_{\overline{h}}\}$, and α is a hyperparameter for adjusting the confidence of background score, which empirically is set to 1. $\boldsymbol{Y}$ is firstly normalized by $\boldsymbol{Y}^c(x,y) = \boldsymbol{Y}^c(x,y)/max_{x,y}\boldsymbol{Y}^c(x,y), c \in [1, C-1]$, and all scores irrelevant to ground truth are thresholding to 0. Finally we concatenate $\boldsymbol{Y}^{c=0}$ into $\boldsymbol{Y}^c$. During inference, $\widetilde{\boldsymbol{Y}}_o$ is used for segmentation, and $\widetilde{\boldsymbol{Y}}_o^{c=0}(x,y)$ is set to a fixed value β.

In all, the CPN is optimized by the final loss function $\mathcal{L}_{all}$ (17), and empirically we set $w_1 = w_2 = w_3 = 1$. Fig. 2 demonstrates the overall CPN framework.

$$\mathcal{L}_{all} = w_1\mathcal{L}_{cls} + w_2\mathcal{L}_{tcp} + w_3\mathcal{L}_{cpcr}. \tag{17}$$

4. Experiments

4.1. Implementation Details

Dataset and evaluated metric: The proposed approach is evaluated on the PASCAL VOC 2012 segmentation benchmark [13]. There are 20 foreground object categories and 1 background annotated in the dataset. Following conventions, the number of training images is 10,582. The validation dataset contains 1,449 images and the test one has 1,456 samples. During the whole training process, only image-level annotation is provided. To measure the performance of all experiments, the mean Intersection-over-Union (mIoU) is used as the evaluation metric.

Network settings: We adopt the ResNet38 [39], as one of the prevailing models in most WSSS frameworks, as the backbone of the CPN. The parameters trained on the ImageNet [11] are used for the initialization of the CPN. Following the previous work, we remove the final GAP layer and fully-connected layer, and replace the last three convolution layers with the atrous convolutions with the adapted dilation rates, so that the output stride of the net is 8. According to [35], for the aggregated features $\boldsymbol{X}$ in PCM and PRCM, we firstly extract feature maps from stage 3 and 4, and then jointly decrease their channels into 64 and 128 by use of 1×1 convolution layers. Lastly, we concatenate these features and input images to form $\boldsymbol{X}$.

Training settings: Typical data augmentation on the training set: randomly scaling, color jittering, randomly cropping the images by 448×448, and horizontal flip. The whole model implemented by Pytorch is trained on 1 RTX 3090 GPU with 24 GB memory. We take a mini-batch size of 4 images to train the CPN for 8 epochs. The initial learning rate is 0.01 and decreases by the poly policy with a decay power of 0.9. We leverage SGD Optimizer using weight decay 0.0005 with momentum 0.9. For each mini-batch, we sort the losses in $\mathcal{L}_{cpcr}$ in descending order, and select the top **20%** losses as the hard examples for training (Online Hard Example Mining (OHEM)) to further improve the performance. Similarly with the settings of [35], we block the gradients backpropagation stream from the PCM and PRCM to the network, to avoid the mutual interference of CAMs and the refined CAMs.

4.2. Ablation Studies

In this section, we aim to certify the effectiveness of CP-N. All experimental results are generated from VOC 2012 *train* set. For a fair comparison, the background score β is the value that results in the best mIoU of the pseudo labels. Note that the patch strategy in Tab. 1-4 is Super-pixel Patch with S_N =200.

Improvements on CAM: To improve the performance of the final masks, it is a common way to aggregate prediction maps with different scales. Tab. 1 shows the mIoU of the segments using the baseline CAM, SEAM [35] and our CPN under single- and multi-scale cases. The results show that our CPN presents superior mining ability than the baseline in all different scaling cases. In the multi-scale test, the CPN improves the mIoU over the baseline by nearly **10%**. For SEAM, which is implemented by a siamese network with equivariant regularization, we adopt the hyperparameters that achieve the best performance in [35]. By adding the PRCM, the new SEAM* outperforms the original one in all scale tests. Compared to SEAM, our framework achieves higher performance (**57.43%**) in the multi-scale test.

Fig. 5 shows several samples of the visualized CAM made by the baseline, SEAM and CPN. Compared to the baseline and SEAM, our CPN can help CAM seek more seeds in low response areas to generate a complete CAM for the foreground. However, for small objects (the last column in Fig. 5), it can be seen that the foreground seeds by CPN are over-segmented since it is indeed difficult to mine out accurate seeds for small objects without boundary.

Effectiveness of regularization and PRCM: Tab. 2 illustrates the effect of every single module in our approach. Note that for the baseline method only $\boldsymbol{l}_{cls}(s_o)$ is included in $\mathcal{L}_{cls}$ since it lacks the CP Pair. Compared with the baseline, the $\mathcal{L}_{tcp}$ and PCM improves the mIoU up to 51.08%. Benefit by $\mathcal{L}_{cpcr}$, the model further achieves a 4.63% improvement. By further applying OHEM to the $\mathcal{L}_{cpcr}$, the results achieves 56.58% mIoU on the VOC12 *train* set. Finally, the model achieves a 0.85% improvement after adopting the PRCM.

Method	Image scale				
	0.5	1.0	1.5	2.0	all
baseline	41.15	48.29	49.51	47.44	47.84
SEAM [35]	49.35	51.57	52.25	49.79	55.41
SEAM*	**49.64**	52.15	53.14	50.55	55.71
CPN	48.91	**54.51**	**55.44**	**54.01**	**57.43**

Table 1. Experiments with single- and multi-scale tests on several methods. * marks the method uses our PRCM module. It is shown that the CPN boosts the overall performance of CAM in various scales, and achieves better CAM than SEAM and the baseline. In addition, our PRCM is effective for improving the results on different scales.

model	mIoU (%)
baseline ($\mathcal{L}_{cls}$)	47.84
baseline + $\mathcal{L}_{tcp}$ + PCM	51.08
baseline+$\mathcal{L}_{tcp}$+ $\mathcal{L}_{cpcr}$+PCM	55.71
baseline+$\mathcal{L}_{tcp}$+ $\mathcal{L}^*{}_{cpcr}$+PCM	56.58
baseline+$\mathcal{L}_{tcp}$+ $\mathcal{L}^*{}_{cpcr}$ + PCM+PRCM	57.43

Table 2. Ablation studies on every part of our method. $\mathcal{L}^*{}_{cpcr}$ refers to $\mathcal{L}_{cpcr}$ with Online Hard Example Mining (OHEM)

Improvements on foreground localization: The CPN aims to improve the CAM by capturing more seeds related to the foreground objects. To verify the idea, we collect the mIoU of background and 20 foreground objects from the baseline, SEAM, and our CPN in Tab. 1. As shown in Tab. 3, compared to the baseline, SEAM achieves compelling mIoU elevation by 5.14% in the background and 7.68% in the foreground. In addition, the CPN improves the mIoU of the foreground up to 56.13%, which outperforms the baseline by 9.75% and the SEAM by 2.07%. The result validates that our CPN can find more foreground object regions than the baseline and the SEAM.

Patch size: Recall from that two patch strategies, namely Grid Patch and Super-pixel Patch, can both be applied on

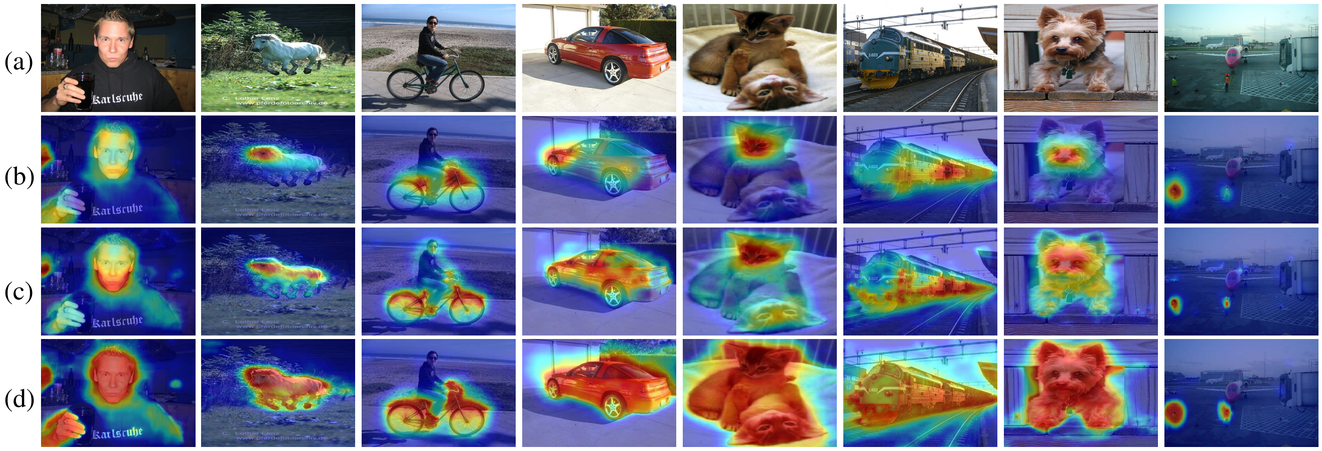

Figure 5. Sample original images (a) and the corresponding visual results by different methods, which are the baseline (b), SEAM (c) and CPN (d). Our method can find more seeds that both the baseline and the SEAM are unable to dig out.

Method	baseline	SEAM	CPN
bg.	77.19	82.33	**83.44**
f.	46.38	54.06	**56.13**

Table 3. The mIoU (%) of the foreground objects (*f.*) and background (*bg.*) in different methods.

our CPN (Sec. 3.2). Note that both patch strategies are closely related to the patch size, and the total number of patches increases with the reduction of it.

For Super-pixel Patch, we explore the effect by simply changing S_N, ranging from 5 to 8000. Fig. 6 reports the mIoU of the results by the CPN with different S_N. It shows that the mIoU firstly shows a rough increasing trend with the increase of S_N, and reaches the peak point (**57.43%**) with $S_N = 200$. Then the quality of CAM declines with higher S_N, and arrives at the lowest point (55.79%) with $S_N = 8000$. Note that lower S_N causes the larger patch size, thus both too high and low patch size in Super-pixel Patch suppress the improvements on the CAM.

For Grid Patch, patch size S is evenly chosen from the set K of some fixed numbers. Thus we implement the experiment by changing the elements of K. We keep $|K| = 2$ and gradually increase the smaller element. Note that the input size of the image is 448. Fig. 7 summaries the results. We notice that with the increase of the S, the mIoU follows a similar tendency to the one in the Super-pixel Patch. The result achieves the best performance (**57.07%**) with $K = \{56, 112\}$, and reaches the bottom with $K = \{4, 7\}$ and $K = \{224, 448\}$, respectively achieving 55.67% and 55.80% mIoU.

Recall that two extreme conditions hold the equality of the CP representation (Sec. 3.1). The results in large and small patch size are exactly corresponding to the condition 1) and 2). Therefore, it is concluded that proper hidden patch size is crucial for significantly improving the performance of the CAM.

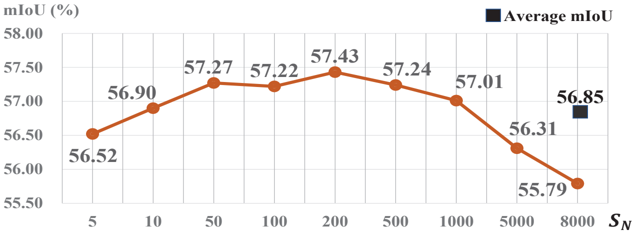

Figure 6. The performance of the Super-pixel Patch strategy with different S_N.

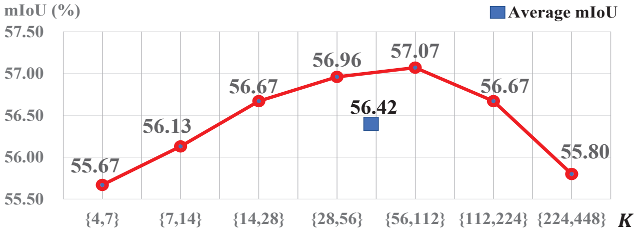

Figure 7. The performance of the Grid Patch strategy with various patch sets K.

Grid Patch vs Super-pixel Patch: To contrast the performance between the two strategies, we respectively calculate the average mIoU of the results in Fig. 7 and Fig. 6. It is observed that CPN with the Super-pixel Patch strategy globally achieves better results (**+0.43%**) than the Grid Patch strategy. It is reasonable since the former one benefits from the pre-classification by the super-pixel. Besides, we test the average time consumption. The Super-pixel Patch (1.45 sec/per image) apparently consumes much more time than the Grid Patch (**0.006 sec/per image**), so the latter one could better meet some real-time operations.

Hidden probability: Recall that the hidden regions in one of the CP Pair are randomly selected with $p_h = 0.5$ (Sec.

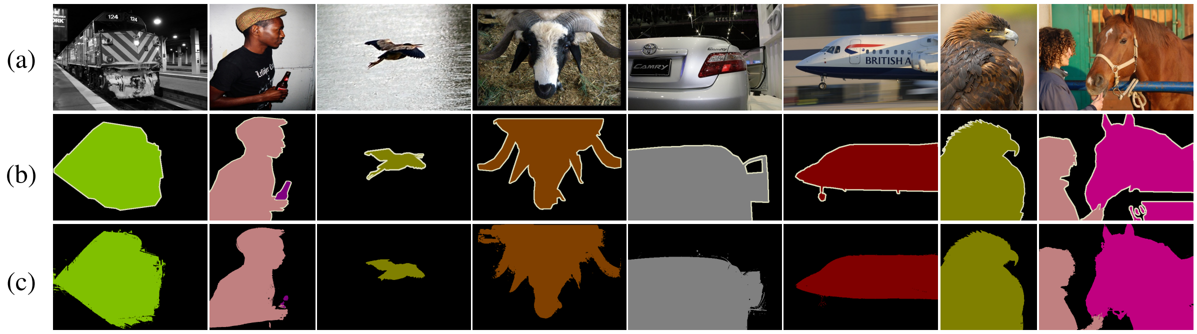

Figure 8. Qualitative results on PASCAL VOC 2012 *val* set. a) Input images. b) Ground-truth labels. c) Our segmentation results (w/ CRF).

3.2). Therefore, the number of the hidden patches in the CP Pair are expectedly equivalent. Here we aim to explore the relationship between the p_h and our CPN. Due to the complementary attribution, we change p_h from 0.1 to 0.5. Tab. 4 shows the CPN with $p_h = 0.5$ achieves the best performance (**57.43%**) , and reaches the bottom (55.52%) with $p_h = 0.1$. The result also validates the effect of the extreme condition 1) on our model since $\boldsymbol{I}_h$ or $\boldsymbol{I}_{\overline{h}}$ is close to $\boldsymbol{I}$ as p_h decreases.

p_h	0.1	0.2	0.3	0.4	0.5
mIoU(%)	55.52	56.87	56.29	57.05	**57.43**

Table 4. The performance of our CPN with different hidden probabilities p_h.

4.3. Comparison with the state of the art

To further improve our CAM, we use a common approach, namely Random Walk (RW) [2], to improve the mIoU of our pseudo labels generated by the CPN up to 67.79%. Following common practice, we then evaluate the quality of the final masks by using DeepLab [26] with ResNet38 backbone. Note that post CRF refinement is used for the output maps. Tab. 5 gives a comparative overview concerning the previous methods. For all the approaches using ResNet38 backbone, our approach presents the state of the art performance on both PASCAL VOC 2012 *val* and *test* set, respectively scoring **67.8%** and **68.5%**. We also note that our results without applying CRF achieve better performance than MCIS [31]. In addition, our method achieves better performance on *test* set than the ICD [14], which uses extra supervision labels. Fig. 8 shows some samples of the final segmentation results, validating the effectiveness of our CPN.

Methods	Pub.	Sup.	Val	Test
*MCOF [34]	CVPR18	$\mathcal{I}+\mathcal{S}$	60.3	61.2
*SeeNet [19]	NIPS18	$\mathcal{I}+\mathcal{S}$	63.1	62.8
*DSRG [20]	CVPR18	$\mathcal{I}+\mathcal{S}$	61.4	63.2
†AffinityNet [2]	CVPR18	$\mathcal{I}$	61.7	63.7
†Single-Stage [3]	CVPR20	$\mathcal{I}$	62.7	64.3
*CIAN [15]	AAAI20	$\mathcal{I}$	64.3	65.3
*FickleNet [24]	CVPR19	$\mathcal{I}+\mathcal{S}$	64.9	65.3
†SSDD [29]	ICCV19	$\mathcal{I}$	64.9	65.5
†SEAM [35]	CVPR20	$\mathcal{I}$	64.5	65.7
*SubCat [5]	CVPR20	$\mathcal{I}$	66.1	65.9
*RRM [41]	AAAI20	$\mathcal{I}$	66.3	66.5
*BES [6]	ECCV20	$\mathcal{I}$	65.7	66.7
†Conta [42]	NIPS20	$\mathcal{I}$	66.1	66.7
*MCIS [31]	ECCV20	$\mathcal{I}$	66.2	66.9
*ICD [14]	CVPR20	$\mathcal{I}+\mathcal{S}$	67.8	68.0
†Ours (w/o CRF)	-	$\mathcal{I}$	66.8	67.6
†Ours (w/ CRF)	-	$\mathcal{I}$	**67.8**	**68.5**

Table 5. Comparison with SOTA on VOC 2012 *val* and *test* in terms of mIoU (%). Methods marked by * use ResNet101 backbone, the others marked by † use ResNet38. The supervision (Sup.) contains image-level label ($\mathcal{I}$) and saliency maps ($\mathcal{S}$).

5. Conclusion

In this paper, we have proposed a simple yet effective pipeline for weakly supervised semantic segmentation with only image-level labels provided. First, from the information theory perspective, we showed that the sum of CAMs generated by a pair of images with Complementary Patch regions (CP Pair) is able to mine out more foreground seeds. Then, based on this observation, we presented a CP Network (CPN) with a bunch of regularization to achieve an improved CAM. To further refine the results, we designed a Pixel-Region Correlation Module (PRCM) to bring more contextual information for the CAM. Extensive experiments on the PASCAL VOC 2012 dataset show that our proposed CPN achieves new state-of-the-art performance.

Acknowledgements: This research was supported in part by National Natural Science Foundation of China (61871325, 62001394), National Key Research and Development Program of China (2018AAA0102803, 2018YFB1703201, 2019YFB1704003, 2019YFB1706602), Shanghai Science and Technology Innovation Action Plan (19511105900), Chinese Ministry of Education Research Found on Intelligent Manufacturing (MCM20180703).

References

[1] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2209–2218, 2019.

[2] Jiwoon Ahn and Suha Kwak. 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 (CVPR)*, pages 4981–4990, 2018.

[3] Nikita Araslanov and Stefan Roth. Single-stage semantic segmentation from image labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4253–4262, 2020.

[4] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *European Conference on Computer Vision (ECCV)*, pages 549–565, 2016.

[5] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8991–9000, 2020.

[6] Liyi Chen, Weiwei Wu, Chenchen Fu, Xiao Han, and Yuntao Zhang. Weakly supervised semantic segmentation with boundary exploration. In *European Conference on Computer Vision (ECCV)*, pages 347–362, 2020.

[7] 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 Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(4):834–848, 2018.

[8] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 801–818, 2018.

[9] Junsuk Choe, Seungho Lee, and Hyunjung Shim. Attention-based dropout layer for weakly supervised single object localization and semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 1–1, 2020.

[10] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1635–1643, 2015.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition (CVPR)*, pages 248–255, 2009.

[12] Thibaut Durand, Taylor Mordan, Nicolas Thome, and Matthieu Cord. Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 642–651, 2017.

[13] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision (IJCV)*, 88(2):303–338, 2010.

[14] Junsong Fan, Zhaoxiang Zhang, Chunfeng Song, and Tieniu Tan. Learning integral objects with intra-class discriminator for weakly-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4283–4292, 2020.

[15] Junsong Fan, Zhaoxiang Zhang, Tieniu Tan, Chunfeng Song, and Jun Xiao. Cian: Cross-image affinity net for weakly supervised semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 10762–10769, 2020.

[16] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision (IJCV)*, 59(2):167–181, 2004.

[17] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3146–3154, 2019.

[18] Seunghoon Hong, Donghun Yeo, Suha Kwak, Honglak Lee, and Bohyung Han. Weakly supervised semantic segmentation using web-crawled videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7322–7330, 2017.

[19] Qibin Hou, PengTao Jiang, Yunchao Wei, and Ming-Ming Cheng. Self-erasing network for integral object attention. In *Advances in Neural Information Processing Systems (NIPS)*, volume 31, pages 549–559, 2018.

[20] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7014–7023, 2018.

[21] Anna Khoreva, Rodrigo Benenson, Jan Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 876–885, 2017.

[22] Alexander Kolesnikov and Christoph H. Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *European Conference on Computer Vision (ECCV)*, pages 695–711, 2016.

[23] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3544–3553, 2017.

[24] Jungbeom Lee, Eunji Kim, Sungmin Lee, Jangho Lee, and Sungroh Yoon. Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 5267–5276, 2019.

[25] Jungbeom Lee, Eunji Kim, Sungmin Lee, Jangho Lee, and Sungroh Yoon. Frame-to-frame aggregation of active regions in web videos for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6808–6818, 2019.

[26] Chen Liang-Chieh, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *International Conference on Learning Representations (I-CLR)*, 2015.

[27] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3159–3167, 2016.

[28] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[29] Wataru Shimoda and Keiji Yanai. Self-supervised difference detection for weakly-supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5208–5217, 2019.

[30] Kwak Suha, Hong Seunghoon, and Han Bohyung. Weakly supervised semantic segmentation using superpixel pooling network. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 4111–4117, 2017.

[31] Guolei Sun, Wenguan Wang, Jifeng Dai, and Luc Van Gool. Mining cross-image semantics for weakly supervised semantic segmentation. In *European Conference on Computer Vision (ECCV)*, pages 347–365, 2020.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*, volume 30, pages 5998–6008, 2017.

[33] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7794–7803, 2018.

[34] Xiang Wang, Shaodi You, Xi Li, and Huimin Ma. Weakly-supervised semantic segmentation by iteratively mining common object features. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 1354–1362, 2018.

[35] Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and X-ilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12275–12284, 2020.

[36] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1568–1576, 2017.

[37] Yunchao Wei, Xiaodan Liang, Yunpeng Chen, Xiaohui Shen, Ming-Ming Cheng, Jiashi Feng, Yao Zhao, and Shuicheng Yan. Stc: A simple to complex framework for weakly-supervised semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39(11):2314–2320, 2016.

[38] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7268–7277, 2018.

[39] Zifeng Wu, Chunhua Shen, and Anton van den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognition (PR)*, 90:119 – 133, 2019.

[40] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *European Conference on Computer Vision (ECCV)*, pages 173–190, 2020.

[41] Bingfeng Zhang, Jimin Xiao, Yunchao Wei, Mingjie Sun, and Kaizhu Huang. Reliability does matter: An end-to-end weakly supervised semantic segmentation approach. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 12765–12772, 2020.

[42] Dong Zhang, Hanwang Zhang, Jinhui Tang, Xian-Sheng Hua, and Qianru Sun. Causal intervention for weakly-supervised semantic segmentation. In *Advances in Neural Information Processing Systems (NIPS)*, volume 33, pages 655–666, 2020.

[43] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890, 2017.

[44] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 13001–13008, 2020.

[45] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, 2016.