This CVPR 2020 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Learning Integral Objects with Intra-Class Discriminator for Weakly-Supervised Semantic Segmentation

Junsong Fan[1,2] Zhaoxiang Zhang[1,2,3*] Chunfeng Song[1,2] Tieniu Tan[1,2,3*]
[1] Center for Research on Intelligent Perception and Computing (CRIPAC),
National Laboratory of Pattern Recognition (NLPR),
Institute of Automation, Chinese Academy of Sciences (CASIA)
[2] School of Artificial Intelligence,
University of Chinese Academy of Sciences (UCAS)
[3] Center for Excellence in Brain Science and Intelligence Technology, CAS
{fanjunsong2016, zhaoxiang.zhang}@ia.ac.cn, {chunfeng.song, tnt}@nlpr.ia.ac.cn

Abstract

Image-level weakly-supervised semantic segmentation (WSSS) aims at learning semantic segmentation by adopting only image class labels. Existing approaches generally rely on class activation maps (CAM) to generate pseudo-masks and then train segmentation models. The main difficulty is that the CAM estimate only covers partial foreground objects. In this paper, we argue that the critical factor preventing to obtain the full object mask is the classification boundary mismatch problem in applying the CAM to WSSS. Because the CAM is optimized by the classification task, it focuses on the discrimination across different image-level classes. However, the WSSS requires to distinguish pixels sharing the same image-level class to separate them into the foreground and the background. To alleviate this contradiction, we propose an efficient end-to-end Intra-Class Discriminator (ICD) framework, which learns intra-class boundaries to help separate the foreground and the background within each image-level class. Without bells and whistles, our approach achieves the state-of-the-art performance of image label based WSSS, with mIoU 68.0% on the VOC 2012 semantic segmentation benchmark, demonstrating the effectiveness of the proposed approach.

1. Introduction

Semantic segmentation, which is a foundation of scene understanding, has achieved great progress in recent years [26, 3, 4, 5]. However, it usually requires large-scale datasets with pixel-level annotations for training [9, 25], which is very costly to obtain. To alleviate the burden of pixel-level annotations, researchers proposed weakly-supervised semantic segmentation (WSSS), which only adopts easily obtained coarse labels for training, e.g., image-level labels [20, 40, 1, 38, 37, 15], scribbles [24, 36], and bounding boxes [7, 19, 34]. This paper focuses on the most challenging problem that only adopts image-level class labels for training semantic segmentation models.

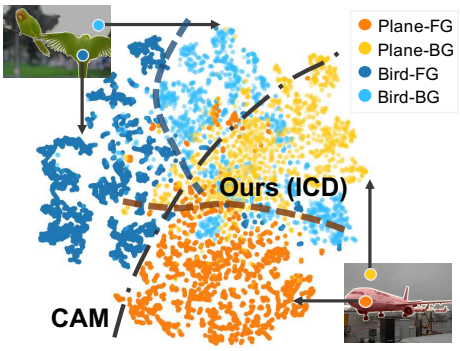

Figure 1. Motivation of the approach. The CAM learns to discriminate between different classes. Our ICD learns to discriminate between the foreground and the background within the same class, which is more suitable for estimating pseudo-masks for the WSSS.

Existing approaches usually follow the pipeline that firstly generates pseudo-masks (a.k.a. seeds) for the target objects and then adopt the seeds to train segmentation models. The class activation map (CAM) [42] is widely adopted to estimate the seeds. However, the CAM can only give the sparse and incomplete estimate of the target object, which is usually the most discriminative region to recognize the object class. Previous approaches try to alleviate this problem by adopting dilated convolutions [40], iterative erasing strategy [38], randomly dropping connections [22], and

*Corresponding Author

online accumulation strategy [18], etc. These approaches achieved good results by forcing the CAM to highlight more unexploited regions. However, due to the intrinsic property that the CAM is only responsible for classification, it is quite tricky to balance the recall of the foreground and the false-positives of the background.

In this paper, we argue that the core problem of applying the CAM to generate seeds is the mismatch between the image-level classification task and the desired pixel-level pseudo-mask estimation task. To train the CAM, pixels vote to obtain the overall score of the image for image-level classification. In this process, the main criterion is the inter-class discrimination. Foreground object pixels in the local regions that are easier to be recognized dominate the activation, e.g., the face of a person or the wheel of a car. Other foreground pixels are overwhelmed and indistinguishable from the background. The ultimate goal of this model is to learn the boundary for inter-class recognition. However, to obtain integral object masks for the WSSS, we need to precisely distinguish whether the pixel belongs to the foreground object or the background. Because the foreground and the background pixels reside in the same image, this discrimination is mainly conducted within the same image-level class, i.e., intra-class discrimination. Generally, the inter-class boundary learned by the CAM does not fit our requirement of intra-class discrimination between the foreground and the background, as illustrated in Fig. 1. Therefore, it is hard to obtain the integral object masks by simply thresholding the CAM score.

To alleviate this boundary mismatch problem, we propose an intra-class discriminator (ICD) that dedicates to separating the foreground and the background pixels within each image-level class, as illustrated in Fig. 1. Such an intra-class discriminator is similar to a binary classifier for each image-level class, which identifies between the foreground pixels and the background pixels. The main difficulty is that we do not have ground truth labels to guide the discriminator. We experimentally observed that the embedded features of the pixels reside in a manifold, and the foreground and the background pixels generally reside in different clusters. Therefore, we leverage this anisotropic property of the features to develop an approach that trains the ICD to separate the foreground pixels from the background, without relying on any additional supervision.

We model the proposed ICD by multiple neural network layers in an end-to-end manner, which can be directly plugged into existing networks, as shown in Fig. 2. Our ICD approach is very efficient and is trained together with the CAM in a single round. The output of the ICD module provides estimates for both the foreground objects and the background clutter. Our approach does not necessarily need any external saliency models [17] to facilitate obtaining background seeds, which are generally required by many recent WSSS approaches [15, 40, 14, 18]. Without referring to saliency models, our approach achieves 64.3% mIoU on the VOC 2012 segmentation benchmark [9], which outperforms many previous approaches with saliency models. Furthermore, with the help of external saliency models for the background estimate, our ICD achieves 68.0% mIoU, which is a new state-of-the-art performance in the image-level label based WSSS field. These results demonstrate the advantage of handling the boundary mismatch problem by our ICD approach.

In summary, the main contributions of our approach are as follows:

- We identify the boundary mismatch problem in applying the CAM to the WSSS, i.e., the gap between image-level inter-class recognition and the desired pixel-level intra-class segmentation.
- We propose an efficient end-to-end Intra-Class Discriminator (ICD) approach to address this problem via learning an intra-class boundary to separate foreground objects and the background.
- We conduct extensive experiments to analyze the effectiveness of our proposed ICD approach. The proposed model achieves the state-of-the-art performance of the image-label based WSSS.

2. Related Work

Class Activation Map. The class activation map (CAM) [42] is widely adopted as the cornerstone of WSSS. The first step to obtain CAM is to train a classification network with the image-level labels, which has a global average pooling (GAP) layer right before the last linear classification layer. Then, it removes the GAP layer and directly applies the classification layer to the feature map, obtaining a dense score map for each class. The grad-CAM [32] is a generalization of the CAM that uses generalized weights to derive the score map, which is also adopted by some WSSS approaches [22]. Because these score maps are derived from classification tasks, they generally only activate on the most discriminative regions for classification, resulting in sparse and incomplete pseudo-masks for the WSSS.

Weakly-Supervised Semantic Segmentation. Weakly-supervised semantic segmentation (WSSS) aims at learning semantic segmentation with only coarse labels, e.g., bounding boxes [7, 34], scribbles [24, 36], and image labels [38, 20, 15, 37, 40, 14]. In this paper, we focus on the most challenging problem that only adopts image-level labels for the WSSS.

Existing image-level label based WSSS approaches usually follow the pipeline that firstly generates pixel-level seeds from image-level labels by the CAM (or grad-CAM),

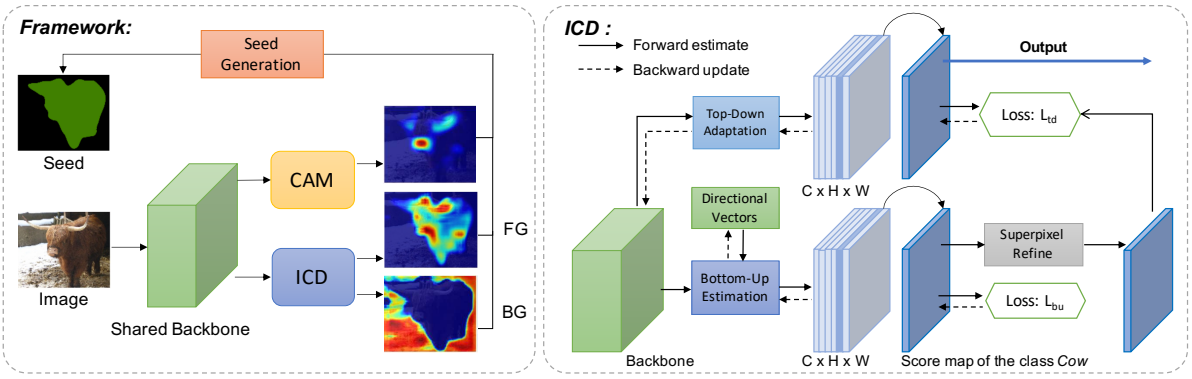

Figure 2. **Left**: The overall framework of our approach, including a branch for CAM and a branch for the proposed ICD. **Right**: The framework of our ICD module, which contains a bottom-up estimation branch and a top-down adaptation branch. The final ICD scores for generating seeds are obtained by the adapted predictions. Please see Sec. 4 for details.

then adopts these seeds to train an ordinary segmentation model. Researchers make great efforts to alleviate the incomplete seed problem. AE-PSL [38] proposes an iterative erasing approach to force the CAM to learn classification from more different regions. MDC [40] proposes to use multiple convolutional layers with different dilation rates to expand the activated regions. DSRG [15] adopts a seed growing algorithm to expand the seeds when training segmentation models. FickleNet [22] proposes to randomly drop connections in each sliding window and accumulate multiple inference results. OAA [18] managed to accumulate the scores map along the training process of CAM. However, because of the intrinsic limitation of the classification task based CAM, it is generally tricky to balance the recall and the false positive. Meanwhile, external saliency models [17, 13] are required to estimate the background, which implicitly introduces additional pixel-level annotation requirements. AffinityNet [1] shows another idea without the requirements of external saliency, which learns the pixels-level affinity model to generate and refine the seeds. However, it needs costly multi-stage training. Besides, the initial seeds still rely on the CAM, which may be inferior because the CAM provides an unreliable estimate of the background.

3. Pilot Study

We conduct pilot experiments to demonstrate that the classification boundary learned by the CAM is inappropriate for separating the foreground and the background for generating seeds. For clarity, we illustrate a two-class case. We apply the trained CAM model to extract features for each pixel in the image. We take these features as individual samples and adopt the t-SNE [27] to visualize them. As shown in Fig. 3(a), the foreground pixels and background pixels are generally located in different clusters and are separable. Fig. 3(b) shows the corresponding CAM score of these pixels, which only highlights partial foreground. Some foreground pixels, e.g., pixels in the red box, though far from the background ones in the manifold, are indistinguishable from the background by the CAM scores, because the CAM only focuses on the boundary between different classes. In contrast, our approach leverages the feature manifold to set a boundary between foreground and background pixels within each class. Fig. 3(c,d) show the scores obtained by our approach, which is more appropriate to generate seeds for the WSSS.

4. Approach

Fig. 2 illustrates the overall framework, which contains a CAM branch to learn the feature manifold from image-level labels, and our ICD branch to learn intra-class boundaries to separate the foreground and the background in each image. The ICD branch contains two main components that estimate the masks based on current features and adapt the whole model for further refinement, respectively.

4.1. Bottom-Up Estimation

The core idea of the ICD is to separate pixels into the foreground and the background groups based on the feature manifold. We tackle this problem by learning a directional vector $\mathbf{w}_c$ for each class c. Let $X = \{X_i\}_{i=1}^N$ be the set of input images, and $\mathbf{f}_i$ be the feature map of X_i, which is of size $H \times W$. The directional vector is obtained by learning:

$$L_{bu}(X) = -\frac{1}{NHW}\sum_{i=1}^{N}\sum_{k=1}^{HW}\sum_{c=1}^{C} y_{i,c}(\mathbf{w}_c^T \mathbf{f}_{i,k})^2 \quad (1)$$

$$\mathbf{w}_c = \frac{\hat{\mathbf{w}}_c}{||\hat{\mathbf{w}}_c||_2} \quad (2)$$

where, $\mathbf{f}_{i,k}$ is the feature of the k-th pixel in $\mathbf{f}_i$, $y_{i,c}$ is the binary label that equals 1 iff the c-th class presents in the

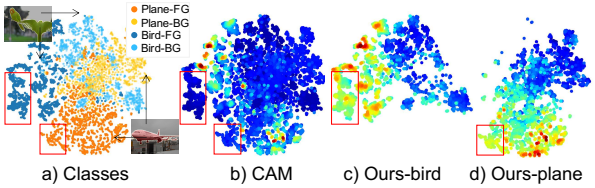

Figure 3. Visualization of the pixels' features via the t-SNE algorithm. a) the class property of the features; b) the CAM scores of the features; c,d) our ICD scores of the features.

image, and C is the total number of foreground classes. To prevent the trivial solution of infinite values, the $\mathbf{w}_c$ is constrained by L2-Normalization as in Eq. 2.

Optimizing Eq. 1 encourages the $\mathbf{w}_c$ to point a direction where pixels' features are located in the two poles. These features share the same image-level class labels and belong to either the foreground of the c-th class or the background. Thus the ICD will not be stuck with the inter-class discrimination problem like the CAM is. Since the foreground (or the background) pixels' features tend to cluster together, and the embeddings of the foreground and the background are generally not overlapped, as illustrated in Fig. 1, the directional vector learns to fit them that one pole is the foreground and another is the background. Thus, it is natural to adopt the sign of the features' projection onto the directional vector to distinguish the foreground and the background. To this end, we compute the score for each pixel by:

$$\hat{S}_{i,k,c} = \mathbf{w}_c^T \mathbf{f}_{i,k} \tag{3}$$

where, the absolute value of $\hat{S}_{i,k,c}$ can be seen as the confidence, and the sign indicates whether the pixel belongs to the foreground or the background. Note that until now, we cannot know if the positive sign stands for the foreground and negative stands for the background or vice versa.

4.2. The ICD Score

In this section, we describe how to adjust the sign of the score in Eq. 3, so that the positive sign always stands for the foreground cluster.

We noticed that though the CAM only activates on partial foreground regions, the highlighted regions concentrate on the foreground. This property is widely demonstrated by the effectiveness of previous methods [15, 40, 14, 18] because they take the high score regions in CAM as the foreground seeds. Therefore, we can employ this property to identify which of the two groups is the foreground. To this end, for each class that presents in the image, we first divide the pixels into two groups according to the sign of Eq. 3, i.e., $I^{pos}_{i,c} = \{k|\hat{S}_{i,k,c} > 0, y_{i,c} = 1\}$ and $I^{neg}_{i,c} = \{k|\hat{S}_{i,k,c} < 0, y_{i,c} = 1\}$. Define $M_{i,k,c}$ as the corresponding CAM score, we compute the average CAM scores for these two groups:

$$\bar{M}^g_c = \frac{1}{N|I^g_{i,c}|} \sum_{i=1}^{N} \sum_{k \in I^g_{i,c}} M_{i,k,c}, \quad g \in \{pos, neg\} \tag{4}$$

where, $|\cdot|$ represents the number of the elements in the set.

Then, we compare the values of $\bar{M}^{pos}_c$ and $\bar{M}^{neg}_c$, and flip the sign of the raw ICD scores if the latter is larger, so that the ICD score $S_{i,k,c}$ always represents the foreground by positive values.

$$S_{i,k,c} = \hat{S}_{i,k,c} \cdot \text{sign}(\bar{M}^{pos}_c - \bar{M}^{neg}_c) \tag{5}$$

where, sign$(\cdot)$ is the Sign function that maps positive values to 1 and negative values to -1.

For efficiency reasons, we implement the above steps as a module in the network, so that the sign of the ICD score can be adjusted online. We compute the local average CAM scores via Eq. 4 in each mini-batch and adopt a moving mean strategy with momentum 0.9 to update the global average CAM scores. Then the Eq. 5 flips the signs according to the global CAM scores.

4.3. Top-Down Adaptation

The previous bottom-up estimation derives the initial foreground and background partitions from existing features. These features are fixed from the view of the directional vectors. To further adapt the model for the pseudo-mask estimation task, we finetune the features by current estimates. To this end, we first refine the ICD scores $S_{i,k,c}$ by averaging their values in each superpixel [10], resulting in refined ICD score $S'_{i,k,c}$. This step is helpful to recover the object boundary information, which generally lost in the downsampling process. Then, we generate the binary mask $B_{i,k,c}$ from the refined ICD score:

$$B_{i,k,c} = \mathbb{I}(S'_{i,k,c} > 0), \quad k \in \{1, ..., HW\} \tag{6}$$

where, $\mathbb{I}(\cdot)$ is the indicator function that equals 1 if the statement is true otherwise 0.

Finally, we adopt a new branch to fit the binary mask and derive new ICD scores $S''_{i,k,c}$:

$$L_{td}(X) = -\frac{1}{NHW} \sum_{i=1}^{N} \sum_{k=1}^{HW} \sum_{c=1}^{C} y_{i,c}(B_{i,k,c} \log \sigma(S''_{i,k,c}) + (1 - B_{i,k,c}) \log(1 - \sigma(S''_{i,k,c}))) \tag{7}$$

where, $\sigma(\cdot)$ is the sigmoid function. $S''_{i,k,c}$ is the branch's prediction, which is also the final adapted ICD score to generate seeds.

4.4. Variants and Analysis

Beside from Eq. 1, there are other methods to learn the directional vector to discriminate the pixels. For example, we can adopt the L1 form to learn the $\mathbf{w}_c$:

$$L_{abs}(X) = -\frac{1}{NHW}\sum_{i=1}^{N}\sum_{k=1}^{HW}\sum_{c=1}^{C} y_{i,c}|\mathbf{w}_c^T\mathbf{f}_{i,k}| \quad (8)$$

Another intuitive choice is to take it as a standard binary classification problem, which generates online pseudo-labels and adopts the sigmoid loss for training:

$$Y_{i,k,c} = \mathbb{I}(\mathbf{w}_c^T\mathbf{f}_{i,k} > 0) \quad (9)$$

$$L_{sig}(X) = \frac{-1}{NHW}\sum_{i=1}^{N}\sum_{k=1}^{HW}\sum_{c=1}^{C} y_{i,c}(Y_{i,k,c}\log\sigma(\mathbf{w}_c^T\mathbf{f}_{i,k}) + (1-Y_{i,k,c})\log(1-\sigma(\mathbf{w}_c^T\mathbf{f}_{i,k}))) \quad (10)$$

We reveal the connections of these variants by analysing the gradients. The gradients of these three variants can be represented in a unified form:

$$\frac{\partial L(X)}{\partial \mathbf{w}_c} = -\sum_{i=1}^{N}\sum_{k=1}^{HW}(y_{i,c}\text{sign}(\mathbf{w}_c^T\mathbf{f}_{i,k}))\cdot\lambda_{i,k}\cdot\mathbf{f}_{i,k} \quad (11)$$

where,

$$\lambda_{i,k} = \begin{cases} |\mathbf{w}_c^T\mathbf{f}_{i,k}| & \text{for} \quad L_{bu} \\ 1 & \text{for} \quad L_{abs} \\ |\sigma(\mathbf{w}_c^T\mathbf{f}_{i,k}) - Y_{i,k,c}| & \text{for} \quad L_{sig} \end{cases} \quad (12)$$

Eq. 11 and 12 show that the gradient is the weighted sum of the features. Our original approach weights more on features with larger absolute projection values, which is generally reliable because they are far from the decision boundary. In contrast, the sigmoid loss approach weights more on features near the boundary, which is inferior because these estimates are not reliable, as demonstrated in Sec. 5.5.

4.5. Training and Generating Seeds

The whole ICD framework is trained together with the CAM. Denote $L_{cam}(X)$ as the multi-class sigmoid loss used by the CAM, the total training loss is:

$$L_{all}(X) = L_{cam}(X) + L_{bu}(X) + L_{td}(X) \quad (13)$$

After training, the adapted ICD score $S''_{i,k,c}$ is adopted to generate seeds. For images of a single class, we directly adopt the ICD scores to generate the pseudo-masks by threshold 0. For images of multiple classes, pixels that are labeled background by all the ICD scores are taken as the background for the seed, and others are foreground. If a pixel is labeled foreground by multiple ICD scores, we adopt the production of the CAM score and the ICD score to determine its class, because the CAM is specifically optimized by the class recognition problem. CRF [21] post-processing is also adopted to refine the details further.

5. Experiments

5.1. Dataset

Following related works, we conduct experiments on the Pascal VOC 2012 [9] to verify our approach. It contains 21 classes (including the background class) for semantic segmentation. There are 10582 training images, which are expanded by [11], 1449 validation images, and 1456 testing images. For all the experiments, we only adopt the image-level class labels for training, which correspond to the 20 foreground classes. The performance is evaluated by the standard mean intersection over union (mIoU) across the 21 classes for the semantic segmentation task.

5.2. Implementation Details

We adopt the VGG16 [33] as the backbone to learn our ICD framework, which is pretrained by the ImageNet [8]. Following the Deeplab's [4] strategy, we change the strides in the last two pooling layers from 2 to 1 to obtain larger feature maps, and we adopt dilation 2 in the Conv5 Block to maintain the receptive field. The Fc6 and Fc7 layers are also changed into fully convolutional layers with 1024 channels and kernel size 3 and 1, respectively.

For the bottom-up estimation, we adopt the Conv5 Block's features to learn. We attach a batch normalization layer [16] with frozen gamma 1 and beta 0 to normalize the features before applying Eq. 1 to prevent trivial single sign results. The directional vector is efficiently implemented by a 1×1 convolution layer. For the top-down adaptation, we attach another Fc6 and Fc7 Block on top of the features that adopted by the bottom-up estimation, and we concatenate these three features to predict the adapted scores. To save the computation burden in the training process, we compute the ImageNet pretrained features for superpixels [10] and follow the strategy in [35] to hierarchically merge them so that each image contains at most 64 superpixels.

New layers are initialized by Normal distribution with a standard deviation 0.01. We use the SGD optimizer with momentum 0.9 and weight decay $5e^{-4}$. The initial learning rate is $1e^{-3}$ and is poly decayed by power 0.9 every epoch, and the learning rate for new layers are multiplied by 10. We adopt the batch size 32 and train 20 epochs. The training images are augmented by random scaling, random flipping, and are randomly cropped into size 321. The bottom-up estimation is learned by single-class images to avoid mixing multi-class objects. We adopt the warm-up strategy that exponentially increases the loss weight from 0 to 1 for the top-down adaptation branch in the first two epochs because initial bottom-up estimates are not reliable.

5.3. Reproductivity

We use two TITAN V GPUs for training, but a single GPU is also feasible. The proposed ICD approach is imple-

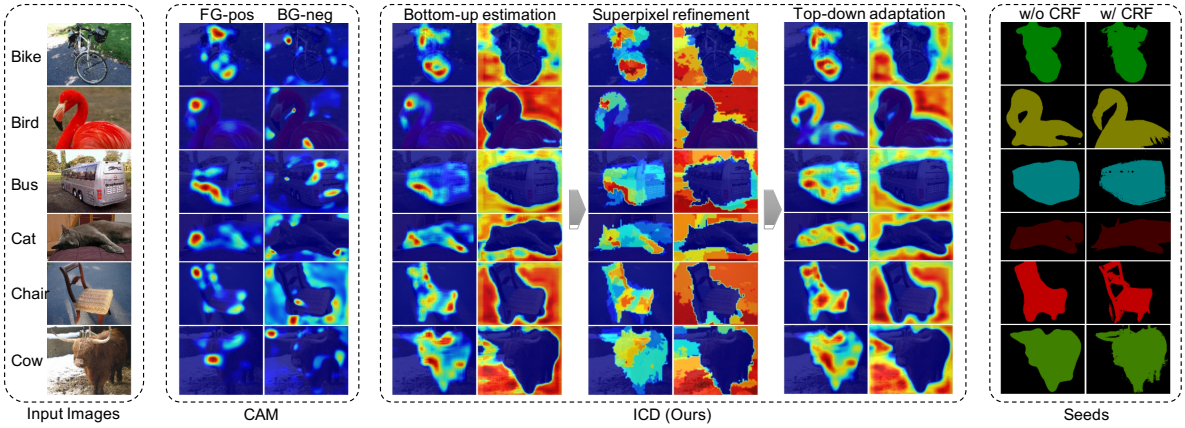

Figure 4. Visualization of the CAM scores, the ICD scores, and the generated seeds. The neighboring two columns of the scores correspond to the foreground and the background, respectively. Best viewed in color.

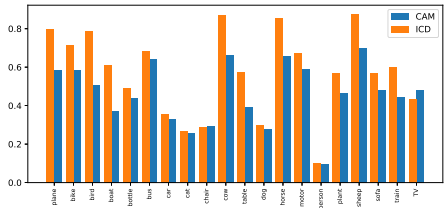

Figure 5. Comparison of the per-class AP of the ICD and the CAM. Results obtained on the VOC12 training set.

Method	mAP (%)
CAM	46.3
Ours (ICD)	**57.0**

Table 1. Evaluation of the quality of score maps by pixel-level mAP on the VOC12 training set. Larger value is better.

Threshold	mIoU (%)		recall (%)	
	CAM	Ours	CAM	Ours
0.0	35.1	59.5	59.7	**83.5**
0.1	53.6	65.3	37.5	78.8
0.3	62.9	72.1	15.1	65.2
0.5	68.5	**75.2**	6.5	43.8

Table 2. Evaluation of the seeds on the VOC12 training set. Higher threshold results in more reliable seeds, but the recall drops.

mented on the MXNet [6] platform. Codes are available at `https://github.com/js-fan/ICD`.

5.4. Evaluating the ICD Score

We first demonstrate the effectiveness of our ICD approach to separating the foreground and the background pixels in the image. Fig. 4 shows some of the examples obtained by the CAM and by our ICD approach. The visualizations are obtained by applying threshold 0 on the score maps. Both the foreground and the background obtained by the CAM are very sparse. What's worse, because the CAM only cares inter-class discrimination, the maxima in the background scores do not cover the surrounding environment; instead, they are often on the edges of the object. This is why many previous approaches [38, 15, 22, 18] rely on additional saliency models to estimate the background. In contrast, our ICD approach dedicates to the discrimination of the pixels sharing the same image class labels, thus derives better estimation than the CAM.

To quantitatively evaluate the above score maps, we recommend applying the mean Average Precision to evaluate the score maps. To compute the AP for a specific class, we first use the scores to rank all the pixels from the images of this class. Then we compute the AP by sequentially taking each pixel as the positive foreground and sampling precisions at all unique recall values. Finally, we compute the mAP by averaging the APs from all the 20 classes. The results are shown in Fig. 5 and Tab. 1. This measure demonstrates that compared with the CAM, our ICD correctly assigns more foreground pixels with higher scores than the background.

We also adopt the final adapted ICD scores to generate seeds to evaluate the quality. We inspect the seeds with multiple different thresholds. Specifically, given the threshold T, the foreground and the background are generated by T and $-T$, respectively. We compute the mIoU with the non-empty regions in the seeds. Generally, a larger threshold results in more reliable estimates, but the recall decreases because of the empty regions. Our ICD seeds consistently outperform the CAM seeds with different thresholds, as shown in Tab. 2.

Bottom-up	Refine	Top-down	CRF	mIoU (%)
✓				49.9
✓	✓			54.0
✓	✓	✓		59.9
✓	✓	✓	✓	62.2

Table 3. Ablation study of the components in our ICD approach. Results are evaluated on the VOC12 training set with the seeds.

5.5. Ablation Study

Components of the ICD. We demonstrate the effect of each component in the ICD framework by evaluating the generated seeds, as shown in Tab. 3. The initial bottom-up ICD score obtains a mIoU 49.9% on the training set by evaluating the generated seeds with the ground truth. The ICD score refined by superpixels achieves seed mIoU 54.0%, demonstrating the superpixel is helpful to revise the ICD scores. The top-down adaptation further boosts the performance by 5.9% mIoU, demonstrating the effect of fine-tuning the backbone model with the pixel-level ICD task. Finally, the CRF post-processing strategy further boosts the performance to mIoU 62.2%, obtaining the full version seeds to train the segmentation models.

Features for the bottom-up estimation. We study the influence of the features on the bottom-up estimation. Specifically, we adopt the features from three different blocks in the VGG16 backbone, i.e., Conv4, Conv5, Fc6, and their concatenation. Results in Tab. 5 show that the bottom-up estimation is sensitive to the features, and the Conv5 performs best. We believe this is because the bottom-up estimation relies on the feature manifolds. The low-level Conv4 features hold too many distractors to estimate the common foreground objects, meanwhile the high-level Fc6 features are over-adapted to the classification task that the features irrelevant to the recognition task are inhibited.

Structures for the top-down adaptation. We study the influence of different structures for the adaptation branch. Specifically, we explore four different structures: **a)** directly adopt the Conv5 Block's features (after BN) for adaptation; **b)** add two blocks, Fc6 and Fc7 that own the same structure as the CAM branch but do not share parameters with it, on top of the Conv5 features, and adopt the Fc7 Block's feature for adaptation; **c)** adopt the concatenation of the above Conv5, Fc6, and Fc7's features for adaptation; **d)** adopt the same structure as c) but share parameters with the CAM branch. The results in Tab. 6 demonstrate that the setting **c)** achieves the best performance. Sharing the parameters with the CAM branch results in inferior results, which reveals that CAM's classification task and our ICD's per-pixel discrimination task require different features. Directly adapt the Conv5 features results in much worse performance. This is because this feature is also directly used by the bottom-up estimation. The guidance for the adaptation is from the bottom-up estimation, thus using the identical feature for these two tasks behaves similarly to update the directional vector and the feature in Eq. 1 simultaneously, which suffers from entangled drifting issues.

Variants	L_{bu}	L_{abs}	L_{sig}
mIoU(%)	**62.2**	58.3	52.9

Table 4. Comparison of the variants of the ICD estimation. Results are evaluated on the VOC12 training set with the seeds.

Method	mIoU (%)	
	w/o CRF	w/ CRF
Conv4	43.8	47.0
Conv5	**49.9**	**56.7**
Fc6	40.0	42.4
Concat	41.3	44.1

Table 5. Comparison of the features for the bottom-up estimation. Results are evaluated on the VOC12 training set with the seeds.

Method	mIoU (%)	
	w/o CRF	w/ CRF
a)	50.4	42.9
b)	57.2	58.8
c)	**59.9**	**62.2**
d)	59.2	62.0

Table 6. Comparison of the structures for the top-down adaptation. Results are evaluated on the VOC12 training set with the seeds.

Variants. We conduct experiments to evaluate variant approaches to estimate the foreground and the background pixels, as discussed in Sec. 4.4. Results in Tab. 4 demonstrate that our approach in Eq. 1 performs best, and the approach adopting the pseudo label and the sigmoid loss performs worst, which is because it weights more on the pixels near the decision boundary, which are not reliable because we lack the ground truth in the weakly supervised scenario.

5.6. Comparison with Related Works

To compare our ICD approach with other related works, we generate seeds with the final adapted ICD scores and refine it with the CRF post-processing algorithm. We use the generated seeds to learn a standard semantic segmentation network. Specifically, we adopt the Deeplab-Largefov, with the VGG16 [33] and the Resnet101 [12] as the backbones. The results are listed in Tab. 7 and Tab. 8, respectively.

Many previous approaches adopt saliency models for generating background seeds, which are usually trained by

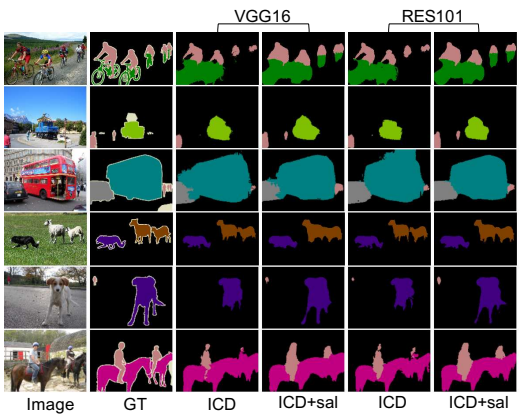

Figure 6. Visualization of the segmentation results. Samples are from the VOC12 val set. We visualize demos both without saliency models (ICD) and with saliency models (ICD + sal).

Method	Supervision	val	test
CCNN [29]$_{\text{ICCV15}}$	I.	35.3	35.6
EM-Adapt [28]$_{\text{ICCV15}}$	I.	38.2	39.6
MIL [30]$_{\text{CVPR15}}$	I.	42.0	40.6
SEC [20]$_{\text{ECCV16}}$	I.	50.7	51.7
AugFeed [31]$_{\text{ECCV16}}$	I.	54.3	55.5
STC [39]$_{\text{PAMI17}}$	I.+S.	49.8	51.2
AE-PSL [38]$_{\text{CVPR17}}$	I.+S.	55.0	55.7
DCSP [2]$_{\text{BMVC17}}$	I.+S.	58.6	59.2
AffinityNet [1]$_{\text{CVPR18}}$	I.	58.4	60.5
GAIN [23]$_{\text{CVPR18}}$	I.+S.	55.3	56.8
MCOF [37]$_{\text{CVPR18}}$	I.+S.	56.2	57.6
DSRG [15]$_{\text{CVPR18}}$	I.+S.	59.0	60.4
MDC [40]$_{\text{CVPR18}}$	I.+S.	60.4	60.8
SeeNet [14]$_{\text{NIPS18}}$	I.+S.	61.1	60.7
FickleNet [22]$_{\text{CVPR19}}$	I.+S.	61.2	61.9
SSNet [41]$_{\text{ICCV19}}$	I.+S.	57.1	58.6
OAA [18]$_{\text{ICCV19}}$	I.+S.	63.1	62.8
Ours	I.	**61.2**	**60.9**
Ours	I.+S.	**64.0**	**63.9**

Table 7. Comparison with related works on the VOC12 dataset. All the results are based on VGG16. I. stands for image-level labels, S. stands for external saliency models.

external saliency datasets with pixel-level annotations and can provide precise background estimates. For a fair comparison, we also evaluate the setting with external saliency models. To this end, we adopt the same saliency model [13] that used by [18] to estimate the background. Specifically, we keep the foreground score unchanged and replace the background score with the saliency scores, then follow the same approach as before to generate seeds.

The results show that our ICD approach outperforms many previous cutting-edge approaches, even without the help of external saliency models. To the best of our knowledge, the previous best VGG16 based result without saliency is achieved by AffinityNet [1] with mIoU 58.4% on the validation set. Our ICD approach significantly promotes this score up to 61.2%, while only using efficient single-stage training. Under the setting of using saliency models, our ICD approach also achieves decent performance, which improves over previous best OAA results with 0.9% mIoU and 2.6% mIoU on the validation set with the VGG16 and the Resnet101 backbones, respectively. We also visualize some of the segmentation models' final predictions in Fig. 6 to help qualitatively evaluate the results. The learned segmentation model correctly handles complicated cases of small objects and multi-objects.

Method	Supervision	val	test
DCSP [2]$_{\text{BMVC17}}$	I.+S.	60.8	61.9
MCOF [37]$_{\text{CVPR18}}$	I.+S.	60.3	61.2
DSRG [15]$_{\text{CVPR18}}$	I.+S.	61.4	63.2
SeeNet [14]$_{\text{NIPS18}}$	I.+S.	63.1	62.8
FickleNet [22]$_{\text{CVPR19}}$	I.+S.	64.9	65.3
OAA [18]$_{\text{ICCV19}}$	I.+S.	65.2	66.4
Ours	I.	**64.1**	**64.3**
Ours	I.+S.	**67.8**	**68.0**

Table 8. Comparison with related works on the VOC12 dataset. All the results are based on ResNet101. I. stands for image-level labels, S. stands for external saliency models.

6. Conclusion

In this paper, we observe the decision boundary mismatch problem in applying the CAM to estimate pseudo-masks for the WSSS. The CAM only learns to discriminate between different classes in the image-level; however, the pseudo-masks require to separate pixels sharing the same class label into the foreground and the background parts. To alleviate this problem, we propose an efficient end-to-end ICD approach, which dedicates to the intra-class discrimination between the foreground and the background pixels in each image. We conduct analysis experiments to study the proposed approach and achieve the new state-of-the-art performance on the VOC 2012 dataset, demonstrating the advantage of the approach.

Acknowledgement

This work was supported in part by the National Key R&D Program of China (No.2018YFB1402600), the National Natural Science Foundation of China (No.61836014, No.61761146004, No.61773375, No.61602481), the Key R&D Program of Shandong Province (Major Scientific and Technological Innovation Project) (NO.2019JZZY010119), and CAS-AIR.

References

[1] 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*, pages 4981–4990, 2018.

[2] Arslan Chaudhry, Puneet K. Dokania, and Philip H. S. Torr. Discovering class-specific pixels for weakly-supervised semantic segmentation. In *British Machine Vision Conference*, 2017.

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

[4] 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*, 40(4):834–848, 2018.

[5] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[6] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *Advances in Neural Information Processing Systems, Workshop on Machine Learning Systems*, 2015.

[7] 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*, pages 1635–1643, 2015.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[9] Mark Everingham, Luc Van Gool, Christopher K.I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

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

[11] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 991–998, 2011.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Proceedings of the European Conference on Computer Vision*, pages 630–645, 2016.

[13] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip HS Torr. Deeply supervised salient object detection with short connections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3203–3212, 2017.

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

[15] 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*, pages 7014–7023, 2018.

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning*, 2015.

[17] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. Salient object detection: A discriminative regional feature integration approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2083–2090, 2013.

[18] Peng-Tao Jiang, Qibin Hou, Yang Cao, Ming-Ming Cheng, Yunchao Wei, and Hong-Kai Xiong. Integral object mining via online attention accumulation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2070–2079, 2019.

[19] 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*, pages 876–885, 2017.

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

[21] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems*, pages 109–117, 2011.

[22] 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 Conference on Computer Vision and Pattern Recognition*, pages 5267–5276, 2019.

[23] Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Tell me where to look: Guided attention inference network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9215–9223, 2018.

[24] 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*, pages 3159–3167, 2016.

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context. In *Proceedings of the*

European Conference on Computer Vision, pages 740–755, 2014.

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

[27] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[28] George Papandreou, Liang-Chieh Chen, Kevin P Murphy, and Alan L. Yuille. Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1742–1750, 2015.

[29] Deepak Pathak, Philipp Krähenbühl, and Trevor Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1796–1804, 2015.

[30] Pedro O Pinheiro and Ronan Collobert. From image-level to pixel-level labeling with convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1713–1721, 2015.

[31] Xiaojuan Qi, Zhengzhe Liu, Jianping Shi, Hengshuang Zhao, and Jiaya Jia. Augmented feedback in semantic segmentation under image level supervision. In *Proceedings of the European Conference on Computer Vision*, pages 90–105, 2016.

[32] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.

[33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[34] Chunfeng Song, Yan Huang, Wanli Ouyang, and Liang Wang. Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3136–3145, 2019.

[35] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013.

[36] Paul Vernaza and Manmohan Chandraker. Learning random-walk label propagation for weakly-supervised semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7158–7166, 2017.

[37] 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*, pages 1354–1362, 2018.

[38] 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*, pages 1568–1576, 2017.

[39] 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*, 39(11):2314–2320, 2017.

[40] 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*, pages 7268–7277, 2018.

[41] Yu Zeng, Yunzhi Zhuge, Huchuan Lu, and Lihe Zhang. Joint learning of saliency detection and weakly supervised semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7223–7233, 2019.

[42] 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*, pages 2921–2929, 2016.