# Object Region Mining with Adversarial Erasing: A Simple Classification to Semantic Segmentation Approach

(2018)

Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, Shuicheng Yan

**Notes**

## 1 Introduction

In a weakly supervised setting, we can either use porposal based methods that are time consuming with the iterative approach, and the classification based methods that provide efficient alternatives, using the classification model to find the most discriminative region in the image and use these region as pseudo labels for semantic segmentation, one problem with such approches is that the model generally only focuses on small and discriminative region in the image (see figure below), but for semantic segmentation we need to obtain masks of the whole elements in the image, to solve this problem the authors propose a form of adversarial erasing, they first train a classification network and use it to localize the most dicriminative regions in the image, and then erase the dicovered region and pass the image through the network to find the new discriminative region, this is done iteratively until no new region is dicovered, with such iterative erasing, the classification network is able to mine other dicriminative region belonging to the object of interest, we can then merge all the regions and use the final mask as pixel wise as targets to train the semantic segmentation network.

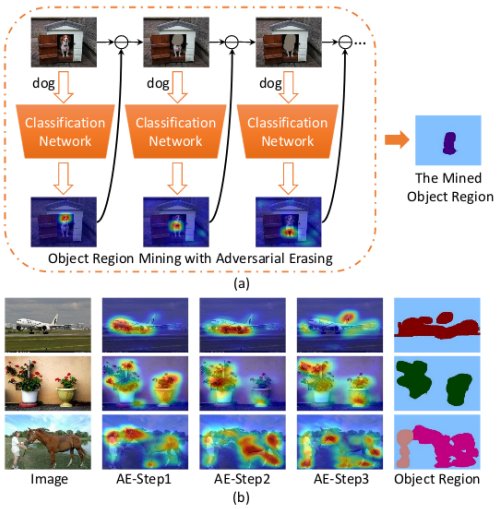


Figure 1. (a) Illustration of the proposed AE approach. With AE, a classification network first mines the most discriminative region for image category label "dog". Then, AE erases the mined region (*head*) from the image and the classification network is re-trained to discover a new object region (*body*) for performing classification without performance drop. We repeat such adversarial erasing process for multiple times and merge the erased regions into an integral foreground segmentation mask. (b) Examples of the discriminative object regions mined by AE at different steps and the obtained foreground segmentation masks in the end.

# 2 Method

## Adversarial Erasing

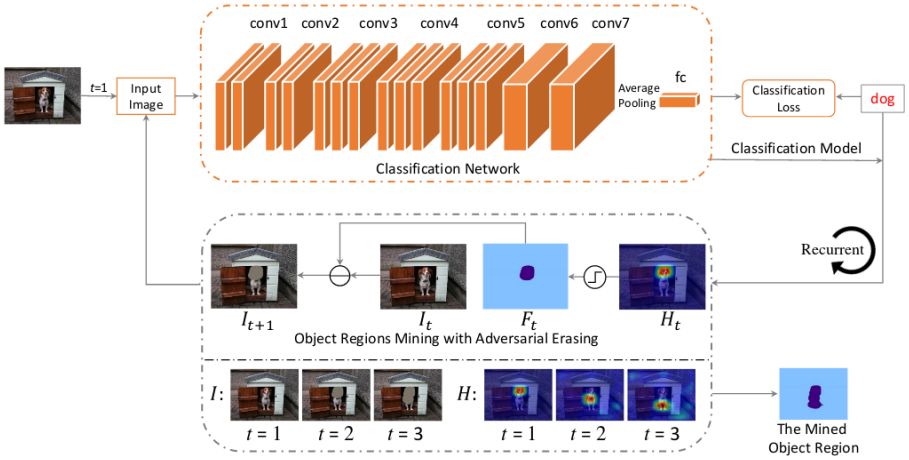


Figure 2. Overview of the proposed adversarial erasing approach. At the step $t$, we first train the classification network with the current processed image $I_t$; then a classification activation method (*e.g.* CAM [34]) is employed to produce the class-specific response heatmap ($H_t$). Applying hard thresholding on the heatmap $H_t$ reveals the discriminative region $F_t$. The proposed approach then erases $F_t$ from $I_t$ and produces $I_{t+1}$. This image is then fed into the classification network for learning to localize a new discriminative region. The learned heatmaps and corresponding proceeded training images with erasing are shown in the bottom. The mined regions from multiple steps together constitute the predicted object regions as output, which is used for training the segmentation network later.

The proposed adversarial erasing is used to find new discriminative regions in an iterative way, this is done in two step: by first training a classification network for localizing the object and then applying adversarial erasing of the dicovered region for more precise segmentation masks, the network is based in the deeplab network using the VGG16, for the classification loss they use MSE loss for multi label classification from *Flexible CNN Framework forMulti-Label Image Classification*:

**MSE loss for multi label classification** Suppose there are $N$ images in the multi-label image set, and $y_i = [y_{i1}, y_{i2}, \ldots, y_{ic}]$ is the label vector of the ith image. $y_{ij} = 1 (j = 1, \ldots, c)$ if the image is annotated with class $j$, and otherwise $y_{ij} = 0$. The ground-truth probability vector of the ith image is defined as $\hat{p}_i = \boldsymbol{y}_i / \|\boldsymbol{y}_i\|_1$ and the predictive probability vector is $p_i = [p_{i1}, p_{i2}, \ldots, p_{ic}]$, And then the cost function to be mini-mized is defined as:

$$J = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{c} (p_{ik} - \hat{p}_{ik})^2$$

At each iteration, the mask to be used for erasing the discriminative parts is obtained using CAM and applying a hard threshold to the heatmap, and then the masked pixel in the original image are replaced by the mean value of the pixels, and the new image is fed to the classification to discover new regions, this is done until the network cannot converge well on the produced training images, this iterative process is done one epoch at the times and not per image.

And this is formally defined in the algorithm bellow:

**Algorithm 1** Object Regions Mining with AE

**Input:** Training data $\mathcal{I} = \{(I_i, \mathcal{O}_i)\}_{i=1}^N$, threshold $\delta$.
**Initialize:** $F_i = \varnothing (i = 1, \cdots, N)$, $t = 1$.
**while** (training of classification is success) **do**
  Train the classification network $M_t$ with $\mathcal{I}$.
  **for** $I_i$ in $\mathcal{I}$ **do**
    **Set** $F_{i,t} = \varnothing$.
    **for** $c$ in $\mathcal{O}_i$ **do**
      Calculate $H_{i,t}^c$ by CAM$(I_{i,t}, M_t, c)$ [34].
      Extract regions $R$ whose corresponding pixel values in $H_{i,t}^c$ are larger than $\delta$.
      Update the mined regions $F_{i,t}^c = F_{i,t}^c \cup R$.
    **end for**
    Update the mined regions $F_i = F_i \cup F_{i,t}$.
    Erase the mined regions from training image $I_{i,t+1} = I_{i,t} \backslash F_{i,t}$.
  **end for**
  $t = t + 1$.
**end while**
**Output:** $\mathcal{F} = \{F_i\}_{i=1}^N$

So in summary, we first train the classification network for multi-class classification, and then we go through each image in sucession, get the class scores, use CAM to get the heat maps, set a threshold to get a mask, add the given region to the region detected in the previous iteration both per class (only the values above the threshold) and them the whole mask, and then erase the last detected region from the image, and do the same in the next iteration.

And to mask the background pixels, they use silency detection, and the location with low silency are considered as background.

### Online PSL (Prohibitive Segmentation Learning) for Semantic Segmentation

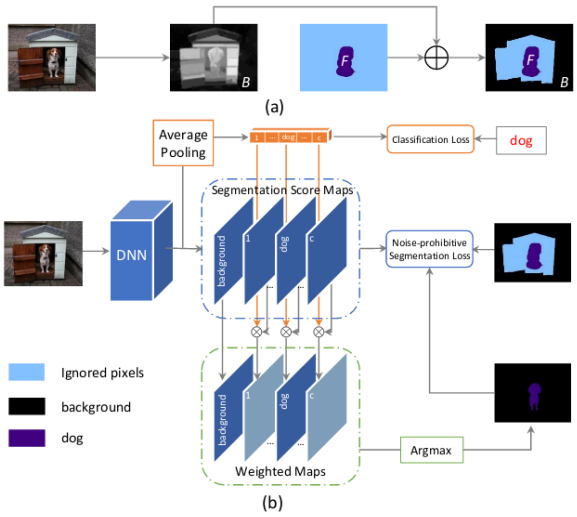


Figure 3. (a) The process of segmentation mask generation. (b) The proposed online PSL approach for semantic segmentation. The classification scores are used to weight "Segmentation Score Maps" to produce "Weighted Maps" in an online manner. Those classes with low classification confidences are prohibited for producing the segmentation mask. Then, both the mined mask and the online produced mask are used to optimize the network.

Even with the iterative adversarial erasing, some object of background related pixels may be missed or they may be noisy, for this they add PSL module for segmentation to further refine the prediction, first we take the vector of classification scores after global average pooling and we use it to weight the prediction per class mask by the deep lab network, this gives us two inputs $S$ wich is the unweighted output and the weighted predictions $\hat{S}$, and we then use these two and compare them to the mask generated by the adversatial erasing (AE), and the PSL aobjective is two minimize both the CE losses of the weighted and non weighted prediction and the AE output:

$$\min_\theta \sum_{I\in\mathcal{I}} J(f(I;\theta),S) + J(f(I;\theta),\hat{S})$$
where
$$J(f(I;\theta),S) = -\frac{1}{\sum_{c\in\mathcal{O}^{seg}}|S_c|}\sum_{c\in\mathcal{O}\,\text{seg}}\sum_{u\in S_c}\log f_{u,c}(I;\theta)$$
and
$$J(f(I;\theta),\hat{S}) = -\frac{1}{\sum_{c\in\mathcal{O}\,\text{seg}}|\hat{S}_c|}\sum_{c\in\mathcal{O}^{seg}}\sum_{u\in\hat{S}_c}\log f_{u,c}(I;\theta)$$

## 3 Experiments

They use DeepLab-CRF-LargeFOV as the base for the classification and segmentation network, 321321 randomly croped images, a learning rate of 0.001, divided by 10 after 6 epochs, and train the whole network for 15 epochs.

Table 1. Comparison of weakly-supervised semantic segmentation methods on VOC 2012 *val* set.

| Methods | Training Set | mIoU |
|---|---|---|
| Supervision: Scribbles | | |
| Scribblesup (CVPR 2016) [12] | 10K | 63.1 |
| Supervision: Box | | |
| WSSL (ICCV 2015) [16] | 10K | 60.6 |
| BoxSup (ICCV 2015) | 10K | 62.0 |
| Supervision: Spot | | |
| 1 Point (ECCV 2016) [22] | 10K | 46.1 |
| Scribblesup (CVPR 2016) [12] | 10K | 51.6 |
| Supervision: Image-level Labels | | |
| (* indicates methods implicitly use pixel-level supervision) | | |
| SN_B* (PR 2016) [28] | 10K | 41.9 |
| MIL-seg* (CVPR 2015) [19] | 700K | 42.0 |
| TransferNet* (CVPR 2016) [7] | 70K | 52.1 |
| AF-MCG* (ECCV 2016) [20] | 10K | 54.3 |
| Supervision: Image-level Labels | | |
| MIL-FCN (ICLR 2015) [18] | 10K | 25.7 |
| CCNN (ICCV 2015) [17] | 10K | 35.3 |
| MIL-sppxl (CVPR 2015) [19] | 700K | 36.6 |
| MIL-bb (CVPR 2015) [19] | 700K | 37.8 |
| EM-Adapt (ICCV 2015) [16] | 10K | 38.2 |
| DCSM (ECCV 2016) [24] | 10K | 44.1 |
| BFBP (ECCV 2016) [23] | 10K | 46.6 |
| STC (PAMI 2016) [29] | 50K | 49.8 |
| SEC (ECCV 2016) [10] | 10K | 50.7 |
| AF-SS (ECCV 2016) [20] | 10K | 52.6 |
| Supervision: Image-level Labels | | |
| AE-PSL (ours) | 10K | **55.0** |

Table 2. Comparison of weakly-supervised semantic segmentation methods on VOC 2012 *test* set.

| Methods | Training Set | mIoU |
|---|---|---|
| Supervision: Box | | |
| WSSL (ICCV 2015) [16] | 10K | 62.2 |
| BoxSup (ICCV 2015) [3] | 10K | 64.2 |
| Supervision: Image-level Labels | | |
| (* indicates methods implicitly use pixel-level supervision) | | |
| MIL-seg* (CVPR 2015) [19] | 700K | 40.6 |
| SN_B* (PR 2016) [28] | 10K | 43.2 |
| TransferNet* (CVPR 2016) [7] | 70K | 51.2 |
| AF-MCG* (ECCV 2016) [20] | 10K | 55.5 |
| Supervision: Image-level Labels | | |
| MIL-FCN (ICLR 2015) [18] | 10K | 24.9 |
| CCNN (ICCV 2015) [17] | 10K | 35.6 |
| MIL-sppxl (CVPR 2015) [19] | 700K | 35.8 |
| MIL-bb (CVPR 2015) [19] | 700K | 37.0 |
| EM-Adapt (ICCV 2015) [16] | 10K | 39.6 |
| DCSM (ECCV 2016) [24] | 10K | 45.1 |
| BFBP (ECCV 2016) [23] | 10K | 48.0 |
| STC (PAMI 2016) [29] | 50K | 51.2 |
| SEC (ECCV 2016) [10] | 10K | 51.7 |
| AF-SS (ECCV 2016) [20] | 10K | 52.7 |
| Supervision: Image-level Labels | | |
| AE-PSL (ours) | 10K | **55.7** |