Tell Me Where to Look: Guided Attention Inference Network

(2018)

Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, Yun Fu

Notes

1 Introduction

In some examples, we may have some sort of a distraction that adds a bias toward the detection of a given object in an image when we only supervise using the classificaiton loss, and in a weakly supervised setting where we want to extract the localization maps as pseudo labels for training the semantic segmentation netwok, this gives us very unprecise masks, in such case (figure bellow) the model has no incentive to focus attention only on the foreground class and generalization performance may suffer when the testing data does not have the same correlation (boats out of water):

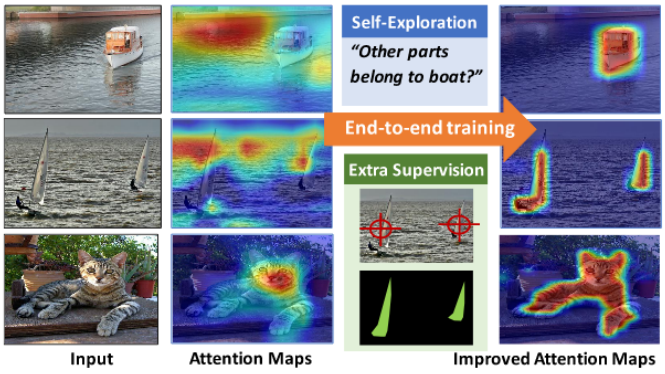

Figure 1. The proposed Guided Attention Inference Network (GAIN) makes the network's attention on-line trainable and can plug in different kinds of supervision directly on attention maps in an end-to-end way. We explore the self-guided supervision from the network itself and propose GAIN$_{ext}$ when extra supervision are available. These guidance can optimize attention maps towards the task of interest.

One solution is to rebalance the dataset to remove such bias, but this is not straightforward, so the authors propose to model the attention maps explicitly as part of the training, with this minimal supervision we are able to control the attention and guide it to the region of interest, so the main contribution of this paper is the guided attention inference networks (GAIN), which aims at supervising attention maps when we train the network for the task of interest.

2 Method

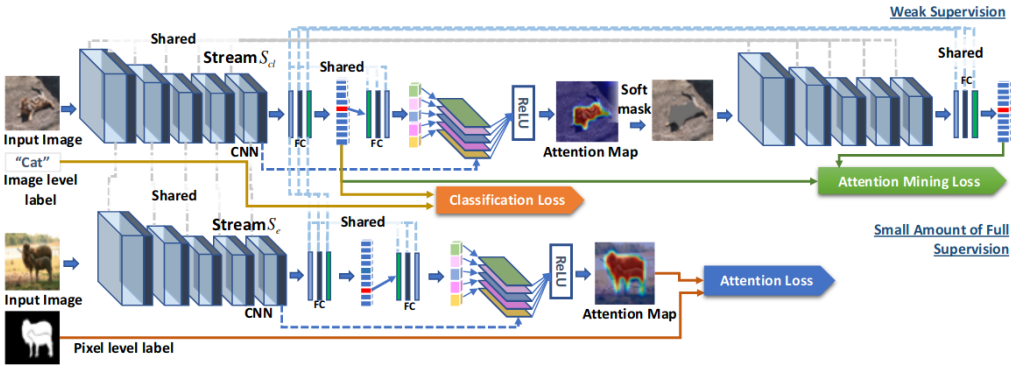

Figure 3. Framework of the GAIN$_{ext}$. Pixel-level annotations are seamlessly integrated into the GAIN framework to provide direct supervision on attention maps optimizing towards the task of semantic segmentation.

To supervise the attention during classification, the authors propose to use two network (that share the same parameters), they call them classification stream S_{cl} and attention manning S_{am}, with the classification stream, we first do the classification and find out the distriminative regions using Grad CAM, given the output scores, we find the derivatives of the score of the correct class (using the ground truth) with respect to the last feature map, this will give us a volume of gradient of the same size as the last feature map, and then we apply global average pooling to get a vector of size # Classes which is the same as the number of channels in the last feature map:

$$w_{l,k}^{c} = \mathrm{GAP}\left(\frac{\partial s^{c}}{\partial f_{l,k}}\right)$$

We then weight the feature map with these weights using a 2D convolution to get the localization maps:

$$A^{c} = \mathrm{Re\,LU}\left(\mathrm{conv}\left(f_{l}, w^{c}\right)\right)$$

And then we can mask the part of the input image where the network did focus to get the classification:

$$I^{*c} = I - \left(T\left(A^{c}\right) \odot I\right)$$

And pass the image through the attention manning S_{am} network to get the class scores, that we'll use to compute the attention manning loss, which is simply the sum of the classification scores that are present in the ground truth of the original image.

$$L_{am} = \frac{1}{n}\sum_{c} s^{c}\left(I^{*c}\right)$$

And the final loss is, where L_{cl} is a sigmoid cross entropy for multi-class classification.

$$L_{\text{self}} = L_{cl} + \alpha L_{am}$$

Extra supervision In case of semi supervised setting with pixel wise labels, we can add a third loss, in the case of an input with a pixel wise labels, we can use the same network to predict the localization maps and consider them as our predicted segmentation masks and compare them to the ground truth masks using MSR:

$$L_{e} = \frac{1}{n}\sum_{c}\left(A^{c} - H^{c}\right)^{2}$$

And this is our final loss in this case: $L_{ext} = L_{cl} + \alpha L_{am} + \omega L_{e}$

And in the end, we use the trained GAIN network to generate pseudo labels for the segmentation network (DeepLab) and train in on these predictions.

3 Experiments

The authors use VGG16 pretrained on image-net, with a batch sier of 1 and a learning rate of 10^{-5}, SGD and train for 35 epochs, for weakly supervised segmentation, they use DeepLab-CRFLargeFOV, and for this they train the network for 8000 iterations with a batch size of 15, initial learning rate of 10^{-3} and decrease by 10 each 2000 iterations.

Methods	Training Set	*val.* (mIoU)	*test* (mIoU)
Supervision: Purely Image-level Labels			
CCNN [19]	10K weak	35.3	35.6
MIL-sppxl [20]	700K weak	35.8	36.6
EM-Adapt [18]	10K weak	38.2	39.6
DCSM [25]	10K weak	44.1	45.1
BFBP [23]	10K weak	46.6	48.0
STC [32]	50K weak	49.8	51.2
AF-SS [21]	10K weak	52.6	52.7
CBTS-cues [22]	10K weak	52.8	53.7
TPL [11]	10K weak	53.1	53.8
AE-PSL [31]	10K weak	55.0	55.7
SEC [12] (baseline)	10K weak	50.7	51.7
GAIN (ours)	10K weak	55.3	56.8
Supervision: Image-level Labels (* Implicitly use pixel-level supervision)			
MIL-seg* [20]	700K weak + 1464 pixel	40.6	42.0
TransferNet* [9]	27K weak + 17K pixel	51.2	52.1
AF-MCG* [21]	10K weak + 1464 pixel	54.3	55.5
GAIN$_{ext}$* (ours)	10K weak + 200 pixel	58.3	59.6
GAIN$_{ext}$* (ours)	10K weak + 1464 pixel	60.5	62.1