LEARNING A WEIGHT MAP FOR WEAKLY-SUPERVISED LOCALIZATION

Tal Shaharabany and Lior Wolf

School of Computer Science, Tel Aviv University

Fig. 1: Given an input image (left), our method trains a generative network to output weights such that a weighted image would be classified similarly to the input image. As training progresses (small panels), the generated weights tend to become more specific. An automatic stopping criterion is used to return the weight map provides good localization.

ABSTRACT

In the weakly supervised localization setting, supervision is given as an image-level label. We propose employing an image classifier $f$ and training a generative network $g$ that outputs, given the input image, a per-pixel weight map that indicates the location of the object within the image. Network $g$ is trained by minimizing the discrepancy between the output of the classifier $f$ on the original image and its output given the same image weighted by the output of $g$. Our results indicate that the method outperforms existing localization methods on the challenging fine-grained classification datasets.

1. INTRODUCTION

Most weakly supervised localization methods assume that an image classifier relies on image regions within the foreground segment, and try to analyze the behavior of the classifier to extract this information [1]. Breaking down this assumption, one can identify three challenges: first, the classifier may rely on context, i.e., on regions outside the object. Second, the classifier may rely on object parts and ignore much of it. Third, the problem of the classifier explainability.

Regarding the first problem, one can attempt to rely on explainability methods that differentiate between positive and negative contributions [2] or assume, as we do, that modern classifiers are less prone to such issues, especially when delineating between similar classes, which tend to appear in similar contexts. The second challenge, which may lead to the identification of parts instead of the entire object, is a major issue in modern weakly supervised localization methods [3, 4]. In our case, we propose solving it with an early stopping technique, since our generative method evolves to become increasingly localized during the training process. Our method, in contrast to many weakly supervised approaches, employs a segmentation network $g$ that provides a pixel-wise weight map, given the input image ($I$). Since it is trained on all images, rather than solving one image at a time, it learns general patterns before it learns to identify specific image locations associated with a given class.

An example is shown in Fig. 1, demonstrating an image from the CUB bird dataset. The output of $g$ is shown for consecutive epochs. As can be seen, first the outline of the bird is identified. As training progresses, the network learns to output specific parts of the bird, which are most relevant to distinguishing it from other birds.

The third challenge we have pointed to was the challenge of extracting the relevant information from the classifier $f$. While many methods rely on the activations of the network and on its gradients [5], this is known to be an unreliable process [6]. In fact, one can already point to the collinearity problem in linear regression as presenting ambiguity regarding the causal links between input and prediction [7].

Our approach, instead of trying to analyze the behavior of the classifier $f$ given $I$ by considering the information flow within $f$, derives a training loss for training network $g$ that only considers the output of $f$. Specifically, we require that
the pseudo-probabilities provided by $f$ given the input image
be the same as the vector of pseudo-probabilities $f$ outputs
on the image $I \cdot g(I)$, which is a pixel-wise weighting of an
image by the weight mask produced by $g$.

Our experiments, similarly to other recent contributions,
focus on datasets for fine-grained classification, in which the
problem of weakly supervised localization is more challeng-
ing, due to $f$ having to attend to well-localized differences
between the classes. As our results show, using the label-
difference criteria on a pretrained classifier, in order to train a
segmentation network $g$, leads to results that are far superior
to the state-of-the-art methods.

2. RELATED WORK

Supervised deep learning algorithms often require a training
set that relies on extensive human annotation. In order to de-
crease the dependency on human annotation, weakly sup-
ervised approaches have been developed [8]. In this paper, we
focus on using image-level annotation to predict the localiza-
tion of the object bounding box.

Fine-grained recognition datasets have become extremely
popular as benchmarks for weakly supervised object localiza-
tion (WSOL). In fine-grained recognition, annotation is more
challenging, since it requires professional knowledge and do-
main experts. Since fine-grained recognition methods require
specialization to differentiate between similar classes, and
since this specialization is often accompanied by the extrac-
tion of localized features [9, 10], the information extracted
from such classifiers is often less accessible when trying to
segment the entire object.

Many algorithms were proposed for the task of WSOL. The Class
Activation Map (CAM) explainability method [11] and its variants [3, 1] identify the salient pixels that lead to
the classification.

Previous WSOL works have been criticized by [12] for
selecting the best checkpoint and the hyperparameters by con-
sidering the test data. This work also proposes an eval-
uation metric for weakly supervised segmentation, where in-
stead of bounding-box annotation, a foreground-background
mask is given. Unlike conventional segmentation metrics, it
considers multiple thresholds, since tuning the threshold is
challenging in the weakly supervised setting. Very recently,
[13] presented a new method for weighting the feature maps.
Evaluation follows the stringent settings presented by [12],
and weakly-supervised segmentation results are presented on
Oxford-flower102 [14].

3. METHOD

This section introduces our method for weakly supervised
fine-grained object localization. Our method employ an
encoder-decoder network $g$ that maps the input image $I \in R^{3 \times W \times H}$ to a pixel-wise weight mask $M \in R^{W \times H}$.

For the training of network $g$ under WSOL settings, we
offer a novel two-stage algorithm: (1) train a classifier $f : R^{3 \times W \times H} \rightarrow R^d$, with global supervision, where $d$ is the
number of categories in the dataset, and (2) freeze the classi-
 fier weights and employ a classification-invariance loss func-
tion for the training of $g$:

$$\min_g \, D(f(I), f(I \cdot g(I))) + R(g(I)),$$

where $D$ measures the discrepancy between the classifier out-
put and $R$ is a regularizer that minimizes the size of the high-
lighted region produced by the weight mask $M = g(I)$. The
goal of using this loss is to maintain the same classifier repre-
sentation $f$ with and without masking by $g$, see Fig 2. In our
implementation, the cross-entropy loss is used for $D$, and $R$
is the L1 norm.

Without the regularization term $R(g(I))$, the architecture
would converge to a naive solution where the output mask is
fixed at one. With the regularization term, the network is en-
couraged to provide more localized solutions, in which non-
relevant attributes are ignored.

Architecture The encoder of $g$ is a ResNet [15], in which
the receptive field is of size 32 $\times$ 32, i.e., the image is down-
scaled five times. The ResNet contains four residual blocks,
with 64, 128, 256, and 512 output channels, whenever the ar-
chitecture of ResNet 18 or ResNet 34 is used, and 256, 512,
1024, and 2048 output channels for the case of ResNet 50
and ResNet 101. Pre-trained weights, obtained by training on
ImageNet [16], are used at initialization.

The decoder of $g$ consists of five upsampling blocks in or-
der to obtain an output resolution that is identical to the origi-
nal image size. Each block contains two convolutional layers
with a kernel size equal to 3 and zero padding equal to one. In
addition, we use batch normalization after the last convolution
layer before the activation function. The first four layers’ ac-
tivation function is a ReLU, while the last layer’s activation
function is sigmoid. Each layer receives a skip connection
from the encoder that has the same spatial resolution [17].

The backbone of $f$ is a ResNet 18 [15] with four blocks
with a receptive field of size 32 $\times$ 32. The Resnet output vec-
tor’s dimension is 512, as obtained from applying a global av-
erage pooling operator on the last feature map. This backbone
is initialized with pre-trained imagenet parameters.

Setting the bounding box The output of $g$ is map $M$ in the
0 to 1 range, obtained by the sigmoid activation function. In
order to derive a bounding box from this map, we follow the
method of [12]. First, a threshold $\tau = \max(M)/10$ is cal-
culated and values lower than $\tau$ are set to zero, obtaining a map
$M$. We then apply topological structural analysis [18] to the
thresholded map, in order to discover the boundaries of the pro-
posed object in the map. The method produces multiple
proposals for the object contours, and we select the contour
with the largest area. The bounding box is obtained by con-
sidering the bounding box of the selected contour.
Early stopping  The discriminative attributes that enable the classifier \( f \) to distinguish between labels belonging to visually similar categories often rely on well-localized regions of the image. As a result, the loss we propose can lead to over-localization of the object in the image.

To overcome this, we rely on an empirical observation: the pixel weighting network \( g \) becomes increasingly well-localized as training progresses. In other words, \( g \) becomes increasingly specific in the highlighted regions as training progresses. This way, it is able to improve the regularization term without sacrificing the loss that measures the discrepancy between the labels.

To exploit the ability of \( g \) to capture the object at intermediate stages of training, while avoiding the pitfalls of using a well-trained \( g \), we propose using an early stopping procedure. Recall that the bounding box selection algorithm selects one contour. If the image has multiple contours, the bounding box would not contain these. We rely on this signal and select an epoch in which the bounding box still contains most of the weights of the mask \( M \).

Specifically, the stop epoch is selected by considering the average score \( S \) among the images in the validation set. For a single sample, it is computed as:

\[
S = \sum_{x_1=x_1}^{x_2} \sum_{y_1=y_1}^{y_2} \sum_{y=0}^{W} \sum_{x=0}^{H} M_{xy},
\]

\( x_1, x_2, y_1, y_2 \) are the bounding box coordinates for this image, and \( H, W \) are image dimensions. The selected checkpoint has the maximal average score.

The early stopping method selects a checkpoint without any additional supervision. Computationally, it is equivalent to the supervised evaluation on the validation set performed by more-supervised methods [19, 20].

4. EXPERIMENTS

We use the exact same benchmark settings as the baseline methods, using the provided train/test split and evaluation protocol. The datasets supply object localization ground-truth annotation only for the validation and test sets. The supervision signal provided during training is the global class-level annotation of the object. No part-level annotation is used, and we do not compare with methods that rely on such annotation.

Benchmarks  We evaluate our algorithm on the three most popular publicly available fine-grained datasets: (1) CUB-200-2011 [21] (2) Stanford Cars [22] (3) Oxford-flowers [14]. In addition, we test our proposed algorithm on the generic image classification dataset tiny-imagenet [23].

On all datasets, except for Oxford-102, we compute two accuracy scores. For both scores, an intersection over union above 0.5 indicates correct localization. In the GT-known-loc score, we compare the bounding box even if the classifier was mistaken. In the Top1-loc, we only consider the result to be correct if the classifier predicted the correct class. While we claim no contribution to the classifier, we present this score as well, since it is prevalent in the literature.

For the CUB and Oxford datasets, which contain foreground-background masks, we also compute a segmentation metric. Specifically, we use the pixel average precision (PxAP) metric proposed by [12].

Implementation details  While the method has many hyperparameters, listed below, all of these are fairly standard and no attempt was made to tune them in any way. The test data was not used in any way to determine the hyperparameters or the early stopping epoch.

In our method, the input image is first resized to 256\( \times \)256,

| Method       | Backbone     | GT loc[%] | Top1 loc[%] |
|--------------|--------------|-----------|-------------|
| CAM (Zhou, 2016) | VGG-GAP     | 51.24     | 37.50       |
| ADL (Choe, 2019) | VGG-GAP     | 75.41     | 52.36       |
| DANNet (Xue, 2019) | VGG-GAP     | 67.70     | 52.52       |
| MEIL (Mai, 2020) | VGG-GAP     | 73.84     | 57.46       |
| SPA (Pan, 2021) | VGG-GAP     | 77.29     | 60.27       |
| RDAP (Choe, 2021) | VGG-GAP     | 76.46     | 59.35       |
| Ours         | VGG-GAP     | 83.46     | 62.04       |

| Method       | Backbone     | GT loc[%] | Top1 loc[%] |
|--------------|--------------|-----------|-------------|
| CAM (Zhou, 2016) | InceptionV3 | 63.30     | 43.67       |
| DANNet (Xue, 2019) | InceptionV3 | 67.03     | 49.45       |
| I2C (Zhang, 2020) | InceptionV3 | 72.60     | 55.99       |
| GCNet (Lu, 2020) | InceptionV3 | 75.30     | 58.58       |
| ADL (Choe, 2019) | InceptionV3 | 68.80     | 53.04       |
| SPA (Pan, 2021) | InceptionV3 | 72.14     | 53.59       |
| RDAP (Choe, 2021) | InceptionV3 | 82.36     | 65.84       |
| Ours         | InceptionV3 | 82.62     | 65.95       |

| Method       | GT-known loc[%] | GT loc[%] | Top1 loc[%] |
|--------------|----------------|----------|-------------|
| CAM (Zhou, 2016) | 65.2  | 56.8     | 88.9        |
| HaS (Singh, 2017) | 87.4  | 76.6     | 87.6        |
| ADL (Choe, 2019) | 82.8  | 73.8     | 88.9        |
| RDAP (Choe, 2021) | 92.9  | 84.1     | 89.7        |
| Ours         | 96.1  | 84.9     | 87.9        |

| Method       | GT-known loc[%] | Top1 loc[%] |
|--------------|----------------|-------------|
| CAM (Zhou, 2016) | 54.56 | 40.55      |
| infoCAM (Qin, 2019) | 57.79 | 43.34      |
| infoCAM+ (Qin, 2019) | 57.71 | 43.07      |
| Ours         | 60.41 | 44.00      |

**Table 1:** CUB benchmark results for three different backbones - pre-trained on ImageNet. Best results are highlighted in bold, second best are underlined.

**Table 2:** Results for Stanford cars

**Table 3:** Results for Tiny-imagenet.
Table 4: Results for CUB [21] segmentation. PxAP aggregates average precision over many thresholds.

| Method            | PxAP |
|-------------------|------|
| CAM (Zhou, 2016)  | 62.57|
| ART(Singh, 2020)  | 75.45|
| Ours              | **76.70**|

Table 5: Results for Oxford flowers segmentation.

| Method            | PxAP |
|-------------------|------|
| CAM(Zhou, 2016)   | 69.0 |
| HaS(Singh, 2017)  | 63.1 |
| ADL(Choe, 2019)   | 69.8 |
| RDAP(Choe, 2021)  | 71.4 |
| Ours              | **75.6**|

and then a crop is applied to reduce its size to 224×224. During training, cropping is applied randomly, while during validation and testing, we crop the center of the image.

The classifier \( f \) is based on Resnet18 backbone for fine-grained datasets, and Resnet50 for tiny-imagenet. The optimizer is SGD with a batch size of 16, and learning rate of 0.001 for 200 epochs, the weight decay is 1e-4. The scheduler decreases the learning rate by a factor of 10 every 80 epochs. For augmentation, the training procedure employs a resize to a fixed size followed by a random crop, as well as a random horizontal flip. During inference, the algorithm employs resize and central crop. The classifier consists of a single fully-connected layer \( R^k \to R^d \), where \( d \) is number of classes, and \( k \) is the latent space size. Network \( g \) for the fine-grained recognition datasets is trained with the SGD optimizer with a batch size of 128 and an initial learning rate of 0.0001 for 100 epochs. The tiny imagenet model is trained with the same SGD optimizer, a batch size of 128, weight decay 5e-5, an initial learning rate of 0.0001, and for 1000 epochs. We test our method on CUB WSOL with three backbones for \( g \) (i) VGGGAP [11] (ii) InceptionV3 [24] (iii) PreResnet18 [15]. For the Cars and flowers datasets, the backbone of \( g \) is Resnet50-SE with pre-trained Imagenet, in accordance with the baselines. For Tiny-imagenet we used a Resnet50 backbone.

During the training phase, both for \( f \) and \( g \), random horizontal flip with 0.5 probability is applied. All models are trained on a single GeForce RTX 2080Ti Nvidia GPU. This attests to the efficiency of our method, but prevents us from running them on much larger generic classification datasets.

**Results**  
Tab 1 lists the performance of our approach compared to the benchmarks for three backbones on the CUB dataset. Our method obtain state-of-the-art results for each of the backbones. Note that the top-1 localization accuracy is high, even though our classifier is standard and not the leading one (we make no claims regarding the performance of the classifier).

Our method avoid over-/under-segmentation of the object, by selecting the check-point that maximizes \( S \).

For Stanford Car [22], we report the baseline results from [13], who ran all methods cleanly. The results presented in Tab. 2 indicate that our method outperforms all baselines in the GT-known localization by a large margin.

Tab. 3 summarises the results for the Tiny-imagenet dataset. All baseline methods employ the same ResNet50 classifier. Our method outperforms the baseline by a margin that is larger than the difference between the baseline methods. The output of our method, in comparison to the ground truth, is presented in Fig 3. As can be seen, our method can match the ground truth bounding box well and is not overly focused on small discriminative regions. The results that evaluate the weight map obtained from \( g \) as a segmentation map are presented in Tab. 4 and Tab. 5 for CUB and the Oxford flowers, respectively. Evidently, the PxAP score for our method is much higher than for the other methods, by a sizable margin.

**5. CONCLUSIONS**

We present a new WSOL method that relies on a novel and elegant training loss that produces leading segmentation results, and with minimal post-processing also outperform. Unlike methods that rely on explainability, the classifier \( f \) serves as a black-box and only a few assumptions are made regarding its nature. Over-segmentation is prevented by applying an early stopping criterion that does not require any pixel-level labels, without modifying the training of \( f \) in any way.

We note that many of the recent WSOL methods improve the underlying classifier \( f \) in order to better match the needs of localization tasks. For example, Singh et al. [25] cover the discriminative parts during the training of the classifier in order to have it rely on additional regions. For a similar reason, Zhang et al. [3] train two classifiers, where one covers the feature maps of the other.

Not modifying \( f \) has another advantage: in fine-grained datasets, CAM emphasizes discriminate parts of the object and does not capture the entire object [20]. Since, given the classifier, CAM is deterministic, improving the CAM heatmaps would be at the expense of the classifier’s accuracy.

**Acknowledgments**  
This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant ERC CoG 725974).
6. REFERENCES

[1] Zhenyue Qin, Dongwoo Kim, and Tom Gedeon, “Rethinking softmax with cross-entropy: Neural network classifier as mutual information estimator,” ArXiv:1911.10688, 2019. 1, 2

[2] Shir Gur, Ameen Ali, and Lior Wolf, “Visualization of supervised and self-supervised neural networks via attribution guided factorization,” in AAAI, 2021. 1

[3] Xiaolin Zhang, Yunchao Wei, Jiashi Feng, Yi Yang, and Thomas S Huang, “Adversarial complementary learning for weakly supervised object localization,” in CVPR, 2018. 1, 2, 4

[4] Xiaolin Zhang, Yunchao Wei, Guoliang Kang, Yi Yang, and Thomas Huang, “Self-produced guidance for weakly-supervised object localization,” in ECCV, 2018. 1

[5] 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 ICCV, 2017. 1

[6] Yuki M Asano, Christian Rupprecht, and Andrea Vedaldi, “A critical analysis of self-supervision, or what we can learn from a single image,” ArXiv, 2019. 1

[7] Carsten F Dormann, Jane Elith, Sven Bacher, Carsten Buchmann, Gudrun Carl, Gabriel Carré, Jaime R García Marqués, Bernd Gruber, Bruno Lafourcade, Pedro J Leitão, et al., “Collinearity: a review of methods to deal with it and a simulation study evaluating their performance,” Ecography, 2013. 1

[8] Hakan Bilen and Andrea Vedaldi, “Weakly supervised deep detection networks,” in CVPR, 2016. 2

[9] Jianlong Fu et al., “Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition,” in CVPR, 2017. 2

[10] Ze Yang, Tiange Luo, Dong Wang, Zhiqiang Hu, Jun Gao, and Liwei Wang, “Learning to navigate for fine-grained classification,” in ECCV, 2018. 2

[11] Bolei Zhou et al., “Learning deep features for discriminative localization,” in CVPR, 2016. 2, 4

[12] Junsuk Choe, Seong Joon Oh, Seungho Lee, Sanghyuk Chun, Zeynep Akata, and Hyunjung Shim, “Evaluating weakly supervised object localization methods right,” in CVPR, 2020. 2, 3

[13] Junsuk Choe, Dongyoon Han, Sangdoo Yun, Jung-Woo Ha, Seong Joon Oh, and Hyunjung Shim, “Region-based dropout with attention prior for weakly supervised object localization,” Pattern Recognition, 2021. 2, 4

[14] Maria-Elena Nilsback and Andrew Zisserman, “Automated flower classification over a large number of classes,” in ICGV. IEEE, 2008. 2, 3

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” in CVPR, 2016. 2, 4

[16] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al., “ImageNet large scale visual recognition challenge,” International Journal of Computer Vision, 2015. 2

[17] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, “U-net: Convolutional networks for biomedical image segmentation,” in MICCAI. Springer, 2015. 2

[18] Satoshi Suzuki et al., “Topological structural analysis of digitized binary images by border following,” Computer Vision, Graphics, and Image Processing, 1985. 2

[19] Jinheng Xie, Cheng Luo, Xiangping Zhu, Ziqi Jin, Weizeng Lu, and Linlin Shen, “Online refinement of low-level feature based activation map for weakly supervised object localization,” in ICCV, 2021. 3

[20] Xiaolin Zhang et al., “Rethinking localization map: Towards accurate object perception with self-enhancement maps,” ArXiv:2006.05220, 2020. 3, 4

[21] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie, “The caltech-ucsd birds-200-2011 dataset,” 2011. 3, 4

[22] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei, “3d object representations for fine-grained categorization,” in ICCV workshops, 2013, pp. 554–561. 3, 4

[23] Yang Le and Xuan Yang, “Tiny imagenet visual recognition challenge,” CS 231N, 2015. 3

[24] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna, “Rethinking the inception architecture for computer vision,” in CVPR, 2016. 4

[25] Krishna Kumar Singh and Yong Jae Lee, “Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization,” in ICCV. IEEE, 2017. 4