F-CAM: Full Resolution Class Activation Maps via Guided Parametric Upscaling

Soufiane Belharbi[1], Aydin Sarraf[3], Marco Pedersoli[1], Ismail Ben Ayed[1], Luke McCaffrey[2], and Eric Granger[1]

[1] LIVIA, Dept. of Systems Engineering, École de technologie supérieure, Montreal, Canada
[2] Goodman Cancer Research Centre, Dept. of Oncology, McGill University, Montreal, Canada
[3] Ericsson, Global AI Accelerator, Montreal, Canada

soufiane.belharbi.1@ens.etsmtl.ca, luke.mccaffrey@mcgill.ca, aydin.sarraf@ericsson.com,
{marco.pedersoli, ismail.benayed, eric.granger}@etsmtl.ca

Abstract

Class Activation Mapping (CAM) methods have recently gained much attention for weakly-supervised object localization (WSOL) tasks. They allow for CNN visualization and interpretation without training on fully annotated image datasets. CAM methods are typically integrated within off-the-shelf CNN backbones, such as ResNet50. Due to convolution and pooling operations, these backbones yield low resolution CAMs with a down-scaling factor of up to 32, contributing to inaccurate localizations. Interpolation is required to restore full size CAMs, yet it does not consider the statistical properties of objects, such as color and texture, leading to activations with inconsistent boundaries, and inaccurate localizations. As an alternative, we introduce a generic method for parametric upscaling of CAMs that allows constructing accurate full resolution CAMs (F-CAMs). In particular, we propose a trainable decoding architecture that can be connected to any CNN classifier to produce highly accurate CAM localizations. Given an original low resolution CAM, foreground and background pixels are randomly sampled to fine-tune the decoder. Additional priors such as image statistics and size constraints are also considered to expand and refine object boundaries. Extensive experiments[1], over three CNN backbones and six WSOL baselines on the CUB-200-2011 and OpenImages datasets, indicate that our F-CAM method yields a significant improvement in CAM localization accuracy. F-CAM performance is competitive with state-of-art WSOL methods, yet it requires fewer computations during inference.
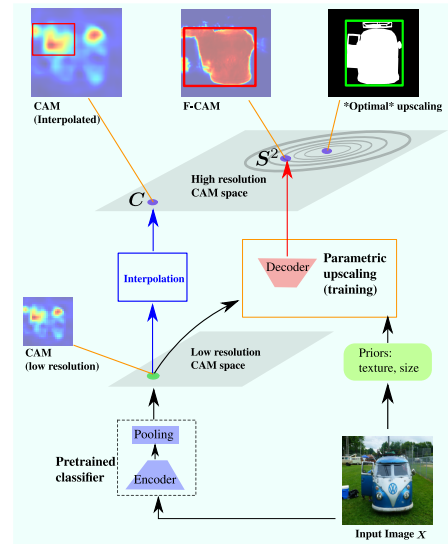
---

[1]Code: https://github.com/sbelharbi/fcam-wsol

Figure 1: An illustration of the differences between interpolation and our trainable parametric upscaling with priors. C is the interpolated CAM, and S^2 is the F-CAM produced using our proposed trainable decoder architecture. The figure also shows the elements required to train our method. More details are presented in Fig. 2.

1. Introduction

Deep learning (DL) models, and in particular CNNs, provide state-of-the-art performance in many visual recognition applications, such as image classification and object detection. However, they remain complex models with millions of parameters that typically require supervised end-to-end training on large annotated datasets. Weakly supervised learning (WSL) has recently emerged as an appealing approach to mitigate the cost and burden of annotating

large datasets, by exploiting data with limited or coarse labels [53]. In particular, WSL is largely beneficial in object localization to avoid costly annotations, such as bounding boxes. Weakly-supervised object localization (WSOL) methods have drawn much attention because they rely on image-class labels, that is less costly to acquire for training.

An important family of WSOL methods includes class-activation maps (CAMs), which visualize CNN decisions by resorting to the feature-map activations of the deep layers [52]. Such feature maps yield spatial information allowing access to a coarse localization of the object. Despite the growing interest and success of CAM methods, they tend to cover only small discriminative parts of an object. Several approaches have been proposed to improve CAMs, such as data-enhancement methods [8, 30, 40, 46, 49] or methods that seek to improve the feature maps [17, 25, 39, 41, 42, 43, 48, 50]. Gradients are also used to interrogate a CNN such that specific target labels are localized [11, 18, 24].

All these methods are typically applied with an off-the-shelf CNN backbone for feature extraction, such as Inception, VGG, or ResNet families. Given the multiple strided convolutions and pooling operations, these backbones yield low resolution CAMs with a downscale factor up to 32^2. Therefore, each pixel in a CAM covers a patch of 32×32 pixels in the input image, making the CAM vulnerable to inaccuracies in object localization. Interpolation is often used to generate full size CAMs, but it does not take into consideration statistical properties of an object such as color and texture or its shape. This results in a well-known issue with CAMs, where they cover only small discriminative image regions, leading to bloby localization with inaccurate boundaries (see Figs. 3 and 4). The downscale factor of CAMs can therefore be a bottleneck in localization tasks[3]. Moreover, CAMs do not explicitly model the background, which plays a central role in increasing false positives/negatives because, e.g., parts of the background could be considered as part of the object [27]. Finally, using only global labels for supervision (without any pixel-level information), is considered to be an ill-posed problem [7, 38] that may lead to sub-optimal solutions.

The issue of low resolution CAMs is not sufficiently addressed in the literature [33, 45, 47]. For instance, [45] proposes dilated residual networks that yield a downscale factor of 8. [47] considers upscaling the feature maps then performing classification and segmentation using two branches. [33] uses a U-Net architecture to reconstruct the image. To obtain full resolution CAMs during inference, the reconstruction image is used in combination with the up-

scaled CAM, in addition to other post-processing methods such as Sobel filtering and region growing methods. Such methods either yield small CAMs, are difficult to scale to large number of classes, or require post-processing steps. In this paper, we explicitly investigate this issue and propose a method to improve the resolution and localization accuracy of CAMs. As an alternative to interpolation, we propose to equip a classifier employed for WSOL with a *parametric* decoder architecture. The decoder is trained to gradually upscale the resolution of feature maps, yielding a full resolution CAM (Fig.2). We explicitly model the foreground and background using the decoder, allowing for robust localization. The decoder outputs two activation maps[4], one for the foreground and the other for the background, with the same size as the input image.

Using a decoder such as in Fig. 2, which has a U-Net form [26], is mainly motivated by deep image prior [36]. It has been successfully applied to super-resolution task and other tasks, including denoising and inpainting [36]. It was shown in [36] that such U-Net architectures with skip connections capture a large part of low-level image statistics. As shown in the recent research in [39], such low-level details play a critical role in improving localization accuracy. The authors of [39] aggregate low-level features to yield detailed CAMs that are also used to collect pseudo-labels for pixel-alignment. This suggests that exploiting fine-grained details in deep networks is a promising direction to improve localization accuracy. Fig.1 shows the intuition of our method and its connection to deep image prior [36]. In
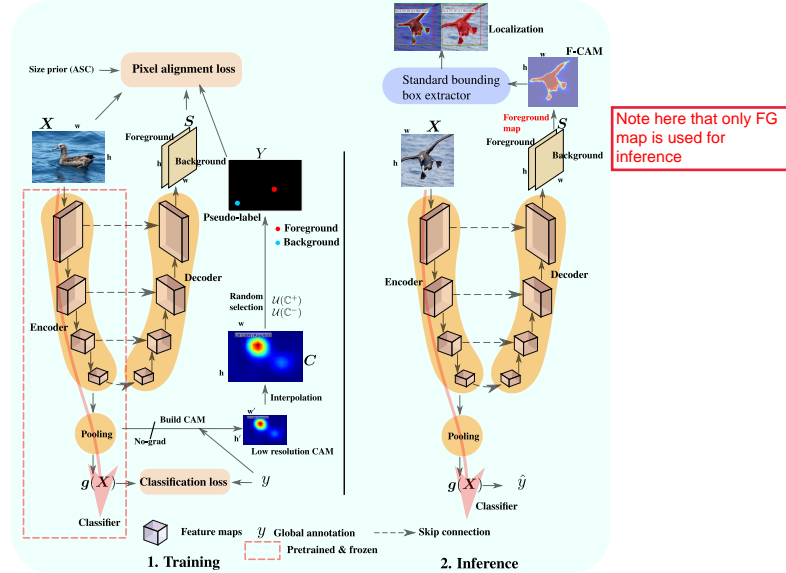

Figure 2: Our proposal: training (left), inference (right).

contrast the transductive learning setup used in [36] for un-

---

[2]In practice, it is common to modify the convolution stride and max-pooling layers to reduce the downsampling factor.

[3]In the supplementary materials, we provide a simulation to show that the downscale factor of a CAM does indeed impose an upper bound performance in localization tasks. We measure the pixel-wise localization with respect to the downscale factor using a simulated CAM.

[4]This could be easily extended to multi-label case.

supervised super-resolution, we consider an inductive learning allowing for fast inference. In particular, we propose using local and global constraints to train the decoder *without additional supervision*. The local constraints entail pixel alignment with pseudo-annotation that are collected from CAMs produced by a WSOL classifier. In addition, conditional random field (CRF) [34] is used as a consistency loss to align the activations with object boundaries, by exploiting statistical properties of the image such as color and local proximity between pixels. An additional global constraint allows recovering the complete part of the object using a size prior.

Our main contributions are summarized below.
(1) We propose a simple, yet efficient alternative to interpolation to improve the quality of CAMs for WSOL task. Given a trained CNN classifier for WSOL, we propose integrating a decoder architecture to perform a parametric upscaling of CAMs. It leverages low/top-level features from the classifier, and original low resolution CAMs to produce full-resolution CAMs (F-CAMs) that yield accurate localizations. In addition, the decoder explicitly models the foreground and background. The proposed guidance loss is named *pixel alignment loss*. **(2)** Training of the decoder is performed using loss terms that enhance localization accuracy, consistency, and completeness of F-CAMs, using only global image-class annotations. We exploit low resolution CAMs in addition to image statistics and object size priors to train the decoder. Our method is generic, and can be combined with any WSOL method with a CNN classifier, such as ResNet, VGG, and Inception families. It aims at improving CAM quality and, therefore, CAM localization accuracy without altering the classification performance. **(3)** Following the experimental WSOL protocol in [7], we combine F-CAM with several baseline methods, including CAM [52], GradCAM [28], GradCam++ [6], Smooth-GradCAM++ [21], XGradCAM [11], LayerCAM [14], on two challenging benchmark datasets for WSOL, CUB-200-2011 [37] and OpenImages [4, 7]. The results indicate that F-CAM can provide significant improvements in CAM localization for these baselines. In addition, F-CAMs produced with our method have better properties than standard CAM methods for WSOL, e.g., better robustness to the threshold values. The enhanced results made these simple WSOL baselines competitive with recent state-of-the-art WSOL methods. Finally, we provide ablation studies and a time complexity analysis. The inference time obtained with F-CAM is lower than the average time of fine-tuned baseline methods, and is competitive with other CAM methods.

2. Related work

Authors in [52] introduce Class Activation Maps (CAMs), showing that spatial feature maps of standard DL models, trained using only image-class labels already, can

rich spatial information which can be used for object localization without additional supervision. CAMs allow highlighting important regions of an input image associated with a CNN's class predictions. Since the CNN is trained for a classification task using global image-class labels, CAMs tend to activate only on small discriminative regions while missing coverage for large parts of the body.

Several extensions have been proposed to alleviate this issue. In particular, WSOL methods based on data enhancement [8, 30, 40, 46, 49] aim to encourage the model to be less dependent on most discriminative regions and seek additional regions. For instance, [30] divides the input image into patches, and only a few of them are randomly selected during training. This forces the model to look for diverse discriminative regions. However, given this random information suppression in the input image, the CNN can easily confuse objects from the background because most discriminative regions were deleted. This leads to high false positives.

Other methods consider improving the feature maps [17, 25, 39, 41, 42, 43, 48, 50]. For instance, [41] considers using dilated convolutions to adapt to objects with different sizes. [48] argues that a WSOL task must be divided into object classification and class-agnostic localization tasks. The latter generates noisy pseudo-annotations, then performs bounding box regression with them for an accurate localization. This is achieved separately from the classification task to avoid undesired interaction between both tasks. In [39], authors enhance the features by considering shallow features of DL models, yielding state-of-the-art localization accuracy, and demonstrating the benefit of shallow features for object localization.

The aforementioned methods are model-dependent, *i.e.* they require training a specific model architecture. Other families of WSOL methods are model-independent, and allow to interrogate the localization of target label over a pre-trained classifier, e.g., Gradient-weighted Class Activation Mapping (Grad-CAM), Grad-CAM++, Ablation-CAM, and Axiom-based Grad-CAM [11, 18, 24]. Dilated residual networks (DRN) [45] explicitly addressed the issue of low resolution of CAMs. Given an input image of 224×224, the DRN can produce a map of size 28×28 which is an improvement compared to ResNet family [13] that can produce a map of 7×7. Despite this improvement, the CAMs are still low resolution with a downscale factor of 8. [33] Uses U-Net model to reconstruct the image. During inference, a series of post-processing steps are used including: applying Sobel filter, and region growing algorithm guided with the upscaled low resolution CAM to yield full resolution CAM. While this provides high resolution CAMs, they still required different post-processing steps.

Finally, authors in [47] use a shared backbone, which is followed by two branches for classification over high res-

olution feature maps and segmentation. CAMs are refined using CRFs, with the collected labels used for segmentation. Our work differs from [47] in terms of the architecture and loss, in addition to the main application. We consider U-Net architecture to exploit low and high-level features of the classifier. The classifier's CAMs are low-resolution, enabling scaling to large numbers of classes. We also exploit an unsupervised size constraint, and other methods to collect reliable regions.

3. Proposed approach

Notation. Consider a training set $\mathbb{D} = \{(\boldsymbol{X}, y)_i\}_{i=1}^N$, where $\boldsymbol{X} : \Omega \subset \mathbb{R}^2$ denotes an image and $y \in \{1, \cdots, K\}$ its global label, with K the number possible classes. Our model (Fig. 2) is composed of: (a) module g for the classification task, and (b) decoder f to output two activation maps, one for the foreground and the other for background, used for the object localization task. The classifier is composed of: (1) feature encoder backbone for constructing features, and (2) pooling head to compute classification scores. For simplicity, $\boldsymbol{\theta}$ refers to the parameters of the entire model (Fig. 2). Furthermore, $g(\boldsymbol{X}) \in [0, 1]^K$ denotes the per-class classification probabilities where $g(\boldsymbol{X})_k = \Pr(k|\boldsymbol{X})$. The softmax activation maps generated by the decoder are denoted $\boldsymbol{S} = f(\boldsymbol{X}) \in [0, 1]^{|\Omega| \times 2}$ where $\boldsymbol{S}^1, \boldsymbol{S}^2$ are the background and foreground maps, respectively. Map $\boldsymbol{S}^2$ is class-agnostic, meaning it can hold the activation of any class y. The classifier yields a low-resolution CAM of the target y, which is then interpolated to have the same resolution as the image. The high-resolution CAM is referred to as $\boldsymbol{C}$. We denote $\boldsymbol{S}_p \in [0, 1]^2$ as a row of matrix $\boldsymbol{S}$, with index $p \in \Omega$ denoting a given point.

Generation of sampling regions (SRs). In order to guide the fine-tuning of the decoder, we employ local information at pixel level as a supervisory signal. A WSOL task aims at producing a bounding box, with the foreground inside and the remaining region considered as background. Similarly, we rely on the activation magnitude in $\boldsymbol{C}$ to decide a pixel state (*i.e.* foreground or background). Since such activations hint the presence or absence of an object, we can assume that pixels with high activations are more likely to be foreground, while lower activations are background [10]. We denote $\mathbb{C}^+$ and $\mathbb{C}^-$ as foreground and background regions, respectively, estimated as follows,

$$\mathbb{C}^+ = \psi^+(\boldsymbol{C}), \quad \mathbb{C}^- = \psi^-(\boldsymbol{C}, n^-), \quad (1)$$

where $\psi^+(\boldsymbol{C})$ is the set of top[5] pixels in $\boldsymbol{C}$, which is ordered from high to low activation. Without adding additional hyper-parameters, $\psi^+(\boldsymbol{C})$ takes all pixels in $\boldsymbol{C}$ with activation magnitude above Otsu [22] threshold obtained

---

[5]In this context, the top n elements of a list ordered from value a toward b are the initial n elements of the list.

over $\boldsymbol{C}$. $\psi^-(\boldsymbol{C}, n^-)$ is the set of top $n^-\%$ pixels in $\boldsymbol{C}$, which is ordered from low to high activation. $\psi^-(\boldsymbol{C}, n^-)$ sorts activation magnitudes in $\boldsymbol{C}$ from low to high and takes the $n^-\%$ of top pixels. We denote by $n^-\%$ the portion of pixels that we are allowed to consider as background. Pixels in $\mathbb{C}^+$ are assigned pseudo-label 1 for foreground and pixels in $\mathbb{C}^-$ are assigned 0 for background.

The estimated sampling regions $\mathbb{C}^+$ and $\mathbb{C}^-$ are uncertain and can contain incorrect labels. Since standard CAMs are often bloby, the foreground $\mathbb{C}^+$ could contain background regions. Similarly, $\mathbb{C}^-$ is expected to hold background pixels but also parts of the object since CAMs are typically incomplete. Due to this uncertainty and noise in labels, we avoid fitting the model directly on $\mathbb{C}^+/\mathbb{C}^-$ all at once. Instead, we randomly select a few pixels at each training iteration, while dropping the remaining pixels [30, 31]. This prevents overfitting over $\mathbb{C}^+$ and $\mathbb{C}^-$. To this end, we define a stochastic set of pixels randomly selected from foreground and background for an image at a training iteration,

$$\Omega' = \mathcal{U}(\mathbb{C}^+) \cup \mathcal{U}(\mathbb{C}^-), \quad (2)$$

where $\mathcal{U}(\mathbb{C})$ consists of a sampled set of one pixel[6] uniformly sampled from the set $\mathbb{C}$. We denote by Y the *partially* pseudo-labeled mask for the sample $\boldsymbol{X}$, where $Y_p \in \{0, 1\}^2$ with labels 0 for background, and 1 for foreground.

Overall training loss. Our training loss combines two main terms: a standard classification loss, and our proposed loss for fine-tuning the decoder named *pixel alignment loss*. This loss entails local (*i.e.*, pixel level) and global terms. The local term aims at aligning the output activations $\boldsymbol{S}$ with the pseudo-labeled pixels selected in Ω' using standard partial cross-entropy $\boldsymbol{H}$. To promote the consistency in activations, and align them with the object boundaries, we exploit statistical properties of the image such as the color, and pixel proximity allowing nearby pixels with similar color to be assigned similar state (*i.e.*, foreground or background). To this end, we include the CRF loss [34] denoted by $\mathcal{R}$. (See supplementary materials for more details.)

CAMs are known to highlight minimal discriminative regions. This could easily lead to unbalanced partitioning, where the background region dominates the foreground in term of size, which gives raise to false positives. To circumvent this issue, the activations of the foreground are explicitly pushed to expand by constraining their total area to be large. In parallel, the background is pushed to be large as well, so as to avoid foreground dominance. By doing so, we push both regions to compete over pixels but without violating other competing losses, *i.e.* CRF and portial cross entropy. In particular, we consider the absolute size constraint (ASC) [3] over both the foreground and background.

---

[6]We can however sample more pixels at once.

We do not assume whether the background is larger than the foreground [23] nor the opposite. The ASC loss encourages both regions to be large. To avoid a trivial solution, with half the image as foreground and the other half as background, control terms are necessary. In this work, partial cross entropy and CRF losses control the growth of the size, so as to ensure consistency with the object boundaries and sampling regions. The size loss is *unsupervised* and formulated through inequality constraints to maximize the area of activation of the map at hand. The constraints are solved via standard log-barrier method [5]. We note: $\boldsymbol{H}(Y_p, \boldsymbol{S}_p) = -\sum_t^2 Y_p^t \log(\boldsymbol{S}_p^t)$ as the standard cross-entropy between the $\boldsymbol{S}_p$ and the pseudo-label mask Y_p at pixel p, and α, λ are balancing coefficients. Our overall pixel alignment loss is formulated as,

$$\min_{\boldsymbol{\theta}} \quad \underbrace{-\log(\boldsymbol{g}(\boldsymbol{X})[y])}_{\text{classification loss (softmax)}} + \alpha \sum_{p \in \Omega'} \boldsymbol{H}(Y_p, \boldsymbol{S}_p) + \lambda \, \mathcal{R}(\boldsymbol{S}, \boldsymbol{X}) ,$$

$$\text{s.t.} \quad \sum \boldsymbol{S}^r \geq 0 , \quad r \in \{1, 2\} , \tag{3}$$

It is important to note that training with our method (Eq. 3) does not require any additional supervision besides the already provided global image-class annotation y. This label is used to build the CAM of the target as presented in Fig. 2. Another important aspect is the semantic meaning of the foreground in $\boldsymbol{S}^2$. Since $\mathbb{C}^+$ holds pixels that are assumed to be foreground estimated from the CAM $\boldsymbol{C}$ of *the true label y,* the foreground predicted in $\boldsymbol{S}^2$ is expected to be consistent with the global annotation y of the image. Once the foreground full-resolution CAM is obtained, localization is carried out using the same standard method used for any CAM (see Fig. 2, *inference*).

4. Results and discussion

4.1. Experimental methodology:

Datasets. To evaluate our method, two datasets from [7] are adopted: CUB-200-2011 (CUB) [37] and OpenImages [4, 7]. CUB contains 200 categories of birds with 5,994 training images and 5,794 testing images. In addition, 1000 extra images annotated in [7] are used as a validation set for model and hyper-parameters selection. OpenImages contains 37,319 images of 100 classes. 29,819 samples are used for training, while 2,500 samples are used for validation. The 5,000 remaining images are used for test. Different from CUB, OpenImages WSOL dataset provides pixel annotation of objects instead of bounding boxes for a fine localization. We follow the protocol in [7] for both datasets. **Evaluation metrics.** Following [7], we report 5 localization metrics and one classification metric. For localization, we report: (1) MaxBoxAcc (also known as CorLoc [9], and GT-known [30]): fraction of images for which the predicted bounding box has more than $\sigma = 50\%$ IoU

with the ground truth, independently from classification prediction, (2) MaxBoxAccV2: the same as MaxBoxAcc but averaged over threes sizes $\sigma \in \{30\%, 50\%, 70\%\}$, (3) top-1 localization accuracy: fraction of images with the correct class prediction and more than $\sigma = 50\%$ IoU with the ground truth box, and (4) top-5 localization accuracy: fraction of images with class label belonging to the top-5 predictions and more than $\sigma = 50\%$ IoU. With OpenImages, (5) we report the PxAP metric proposed in [7] which computes the area under the precision-recall curve. As in [7], the CAM's threshold is marginalized over the interval $\tau \in [0, 1]$ with a step of 0.001.

Implementation details. In all experiments, we follow the same protocol as [7] including backbones, training epochs (50 for CUB and 10 for OpenImages), and batch size of 32. We validated our method over three backbones, VGG16 [29], InceptionV3 [32], and ResNet50 [13]. In Eq.3, the hyper-parameter λ for the CRF is set to the same value as in [34] which is $2e^{-9}$. For log-barrier optimization, hyper-parameter t is set to the same value as in [1, 15]. It is initialized to 1, and increased by a factor of 1.01 in each epoch with a maximum value of 10. α is searched in $\{1, 0.1\}$ through validation. We find that large values, such as 1, do not harm the performance, suggesting that sampling regions are less noisy. In all experiments with our method, we used a learning rate of 0.01 using SGD for optimization. Similar to [7], images are resized to 256×256, then randomly cropped to 224×224 for training. n^- is chosen using the validation set from the set $[0.1, 0.7]$ with a step of 0.1.

Baseline models. To validate our F-CAM method, we compare with recent WSOL methods, including: CAM [52], HaS [30], ACoL [49], SPG [50], ADL [8], CutMix [46], CSTN [20], TS-CAM [12], MEIL [19], DANet [42], SPOL [39], ICL [16], NL-CCAM [44], I²C [51], Grad-CAM [28], GradCam++ [6], Smooth-GradCAM++ [21], XGradCAM [11], LayerCAM [14]. We present the results reported in [7] for the methods: CAM, HaS, ACoL, SPG, ADL, and CutMix. For GradCAM, GradCam++, Smooth-GradCAM, XGradcam, and LayerCAM, we have reproduced their results. We also reproduced results for CAM [52], and name its results as CAM* to distinguish it from CAM's results in [7]. For the rest of the methods, we present what was reported in the original papers. Missing values are shown here by $--$. [7] provides results using few-shot learning (FSL) where a few fully supervised samples are used to train the model. A simple baseline is also provided in [7] which is a center-Gaussian baseline. It generates isotropic Gaussian score maps centered at the image. This represents a lower bound performance obtained without any training. We study the impact of combining our method with 6 baseline WSOL methods over localization performance. Our choice is based on low complexity – we chose methods that yield CAMs by a simple forward

pass, such as CAM [52], or a forward and a backward pass, such as GradCAM family while using standard pretrained classifiers. In addition, these methods simply interrogate a classifier without changing its architecture, making the integration of our decoder with the classifier straightforward. To this end, we select the following WSOL methods: CAM, GradCAM, GradCAM++, Smoth-GradCAM, XGradCAM, and LayerCAM. All the baselines use the same pooling method which is a global average pooling [52]. To integrate our F-CAM method, the WSOL baseline method is trained only using the classification term in Eq.3 until convergence. Then, we freeze the classifier, and continue fine-tuning our decoder using the pixel alignment loss in Eq.3. Such separation in training is meant to avoid any undesirable interaction between classification and pixel-wise assignment tasks [2], and to provide clear conclusions about the results. Moreover, it allows the baseline method to converge and yield accurate localization which will be used to guide the decoder's fine-tuning.

4.2. Comparison with state-of-the-art:

	CUB (MaxBoxAcc)				OpenImages (PxAP)			
Methods	VGG	Inception	ResNet	Mean	VGG	Inception	ResNet	Mean
CAM [52] *(cvpr,2016)*	71.1	62.1	73.2	68.8	58.1	61.4	58.0	59.1
HaS [30] *(iccv,2017)*	76.3	57.7	78.1	70.7	56.9	59.5	58.2	57.8
ACoL [49] *(cvpr,2018)*	72.3	59.6	72.7	68.2	54.7	63.0	57.8	58.4
SPG [50] *(eccv,2018)*	63.7	62.8	71.4	66.0	55.9	62.4	57.7	58.6
ADL [8] *(cvpr,2019)*	75.7	63.4	73.5	70.8	58.3	62.1	54.3	58.2
CutMix [46] *(eccv,2019)*	71.9	65.5	67.8	68.4	58.2	61.7	58.7	59.5
Best WSOL	76.3	65.5	78.1	70.8	58.3	63.0	58.7	59.5
FSL baseline	86.3	94.0	95.8	92.0	61.5	70.3	74.4	68.7
Center baseline	59.7	59.7	59.7	59.7	45.8	45.8	45.8	45.8
CSTN [20] *(icpr,2020)*	Resnet101 [13]: 76.0			–	–	–	–	–
TS-CAM [12] *(corr,2021)*	Deit-S [35]: 83.8			–	–	–	–	–
MEIL [19] *(cvpr,2020)*	73.8	–	–	–	–	–	–	–
DANet [42] *(iccv,2019)*	67.7	67.03	–	–	–	–	–	–
SPOL [39] *(cvpr,2021)*	–	–	96.4	–	–	–	–	–
CAM* [52] *(cvpr,2016)*	61.6	58.8	71.5	63.9	53.0	62.7	56.8	57.5
GradCAM [28] *(iccv,2017)*	69.3	62.3	73.1	68.2	59.6	63.9	60.1	61.2
GradCAM++ [6] *(wacv,2018)*	84.1	63.3	81.9	76.4	60.5	64.0	60.2	61.5
Smooth-GradCAM++ [21] *(corr,2019)*	69.7	66.9	76.3	70.9	52.2	61.7	54.3	56.0
XGradCAM [11] *(bmvc,2020)*	69.3	60.9	72.7	67.6	59.0	63.9	60.2	61.0
LayerCAM [14] *(ieee,2021)*	84.3	66.5	85.2	78.6	59.5	63.5	61.1	61.3
CAM* [52] + ours	87.3	82.0	90.3	86.5	67.8	71.9	72.1	70.6
GradCAM [28] + ours	87.5	84.4	90.5	87.4	68.6	70.0	70.9	69.8
GradCAM++ [52] + ours	91.5	84.6	91.0	89.0	64.8	67.1	66.3	66.0
Smooth-GradCAM++ [52] + ours	89.1	86.8	90.7	88.8	60.3	65.4	64.4	63.3
XGradCAM [52] + ours	86.8	84.4	90.4	88.8	68.7	71.3	70.4	70.1
LayerCAM [52] + ours	91.0	85.3	92.4	89.7	64.3	64.9	65.3	64.8
Best WSOL + ours	91.5	86.8	92.4	89.7	68.7	71.9	72.1	70.6

Table 1: Performance on MaxBoxAcc and PxAP metrics.

Quantitative comparison. Tab.1 shows the performance obtained with the proposed and baseline methods according to the MaxBoxAcc and PxAP metrics. We observe that results with the 6 selected baselines range from the lower performance using CAM* to the higher performance using LayerCAM. This provides a good scenario to evaluate our method when combined with weak and strong baselines. Note that compared to the methods reported in [7], Grad-

	top-1 localization			top-5 localization		
Methods	VGG	Inception	ResNet	VGG	Inception	ResNet
CAM [52] *(cvpr,2016)*	45.8	40.4	56.1	–	–	–
HaS [30] *(iccv,2017)*	55.6	41.1	60.7	–	–	–
ACoL [49] *(cvpr,2018)*	44.8	46.8	57.8	–	–	–
SPG [50] *(eccv,2018)*	42.9	44.9	51.5	–	–	–
ADL [8] *(cvpr,2019)*	39.2	35.2	41.1	–	–	–
CutMix [46] *(eccv,2019)*	47.0	48.3	54.5	–	–	–
ICL [16] *(accv,2020)*	57.5	56.1	56.1	–	–	–
CSTN [20] *(icpr,2020)*	Resnet101 [13]: 49.0		–	–	–	–
TS-CAM [12] *(corr,2021)*	Deit-S [35]: 71.3			Deit-S [35]: 83.8		
I²C [51] *(eccv,2020)*	–	56.0	–	–	68.3	–
MEIL [19] *(cvpr,2020)*	57.4	–	–	–	–	–
DANet [42] *(iccv,2019)*	52.5	49.4	–	61.9	60.4	–
NL-CCAM [44] *(wacv,2020)*	52.4	–	–	65.0	–	–
SPOL [39] *(cvpr,2021)*	–	–	80.1	–	–	93.4
CAM* [52] *(cvpr,2016)*	33.5	40.9	47.8	52.7	54.8	66.5
GradCAM [28] *(iccv,2017)*	18.5	41.6	32.3	41.7	56.2	56.4
GradCAM++ [6] *(wacv,2018)*	20.8	41.4	34.7	47.5	56.8	61.4
Smooth-GradCAM [21] *(corr,2019)*	17.7	44.4	33.1	40.2	60.2	57.7
XGradCAM [11] *(bmvc,2020)*	18.5	40.9	48.3	41.7	55.2	67.3
LayerCAM [14] *(ieee,2021)*	24.8	43.9	44.2	47.8	59.7	70.3
CAM* [52] + ours	44.0	54.6	59.1	70.1	73.8	82.6
GradCAM [28] + ours	22.1	56.1	38.8	50.0	75.4	68.5
GradCAM++ [6] + ours	23.2	55.9	39.0	52.0	76.0	68.9
Smooth-GradCAM [21] + ours	22.9	57.3	38.9	51.2	77.6	68.6
XGradCAM [11] + ours	22.0	55.8	59.3	49.6	75.3	82.7
LayerCAM [14] + ours	23.1	56.4	47.7	51.9	76.6	76.1

Table 2: The top-1 and top-5 localization accuracy of WSOL methods on CUB according to the MaxBoxAcc metric. In red are the cases where our method decreases performance over the corresponding baseline.

CAM family reported much higher performance. Combing our method with each of these baselines yields a considerable improvement for all CAM methods, CNN backbones, and over both datasets. For instance, CAM* alone yields a MaxBoxAcc of 71.5% over CUB using ResNet50. When combined with our method, its performance climbs to 90.3%. The same improvement is observed over OpenImages but with a smaller margin. Note that OpenImages is more challenging than CUB as it has a high variation among classes of different objects (CUB contains only birds' species). Results indicate that by simply combing a decoder with these baselines, we can increase their performance to a level that is competitive with recent state-of-the-art methods, such as TS-CAM and CSTN and even approaching the performance of SPOL. In addition, these results are competitive with FSL, and in some cases, surpassing FSL performance. Similar behavior is observed over MaxBoxAccV2 performance[7].

Tab.2 shows top-1 and top-5 localization performance over CUB. When using our method, WSOL baselines obtained competitive top-5 localization over Inception and ResNet architectures. However, top-1 localiza-

---

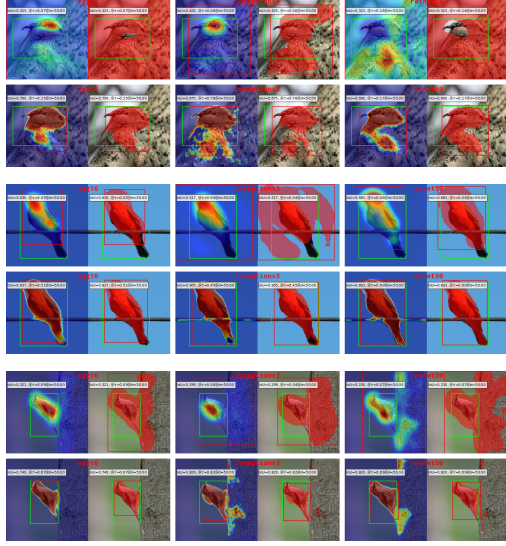[7] MaxBoxAccV2 performance on CUB is reported in supp. material.

Figure 3: Test samples from CUB. Top: CAM*. Middle: GradCAM. Bottom: GradCAM++. First row: WSOL baseline. Next row: WSOL baseline + ours. First column: CAM. Next column: localization. Colors: predicted boxes in red, and true box in green. Thresholded mask is in red. $\sigma = 50\%$.
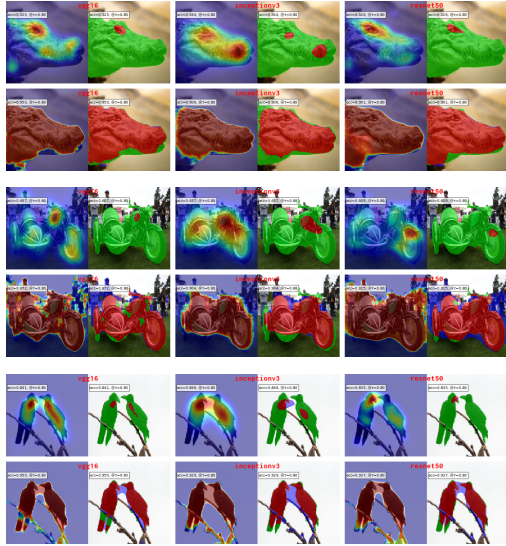

Figure 4: Test samples from OpenImages. Top: Layer-CAM. Middle: Smooth-GradCAM. Bottom: XGradCAM. First row: WSOL baseline. Next row: WSOL baseline + ours. First column: CAM. Next column: localization. Colors: green false negative, red : true positive, blue: false positive. $\tau = 0.8$.

tion is poor even with the assistance of our method. Since `top-1` localization is directly tied to classification performance, the improvement of our method is bounded by

the number of correctly classified samples. By inspecting the classification performance of WSOL baselines, only CAM* yields high accuracy over the CUB dataset for the three architectures. This explains why `top-1` CAM* is much higher than others. The average classification accuracy ranges from 44% to 59%. In addition, all WSOL baselines yield high classification using VGG16 over CUB which again explains the high `top-1` localization. Over OpenImages, all methods yield relatively the same classification accuracy with an average that ranges from 63% to 70%. Since the aim of our method is to improve localization performance without changing the classification performance, our method can only improve the localization of correctly classified samples. This gives advantage to other methods that have high classification accuracy. Note that model selection during the training of our baselines is achieved through the localization performance on the validation set as proposed in [7]. It was observed in [7] that the localization task converges in the early training epochs while the classification task takes longer to converge. We observed similar behavior when training our baselines[8].

Visual comparison: WSOL baselines with F-CAMs. Figs.3 and 4 illustrate the impact of our method on WSOL baselines in term of activations and localization. It is well known that standard CAM methods tend to activate only over minimal discriminative regions as shown in these figures. However, the optimum threshold allows other low (invisible) activations to participate in defining the final localization such as in the first row in Fig.3. As a result, non-discriminative regions could be easily included in the localization. Moreover, such low activations downgrade the interpretability aspect since they yield a box that covers regions without any visible activation over an object. In comparison, our method yields sharp and complete CAMs allowing bounding boxes to be tight around the object making it robust to thresholding while improving the interpretability aspect. Similarly, over OpenImages, WSOL baselines yield minimal activation that are sensitive to thresholding. Note that in `PxAP` metric, authors in [7] do not define an optimum threshold as in `MaxBoxAcc` and `MaxBoxAccV2` since `PxAP` defines an area under the precision-recall curve. Using high threshold such as 0.8 yields very small true positive regions with high false negative. On the other hand, our method easily covers the object with minimal false positives. In the next section, we analyze the score of the CAMs as an attempt to understand their sensitivity to thresholding.
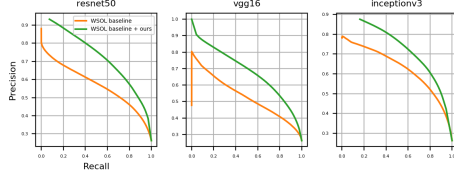
Activation distribution shift: WSOL baselines with F-CAMs. Fig.5 shows the change in `MaxBoxAcc` and `PxAP` metrics with respect to τ over the test set using CAM* base-

---

[8]Training curves and classification performance for all WSOL baselines methods are reported in the supplementary materials.
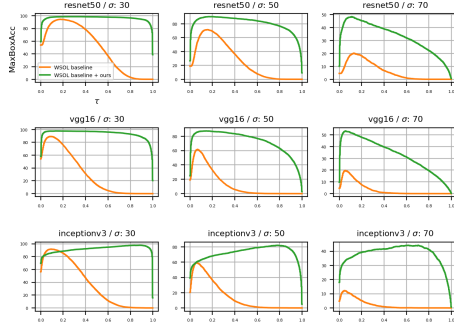
(a) `MaxBoxAcc` metric.

(b) `PxAP` metric.

Figure 5: Localization accuracy as a function of threshold value τ for CAM* (orange) and CAM* + ours (green). (A) CUB. (b) OpenImages.
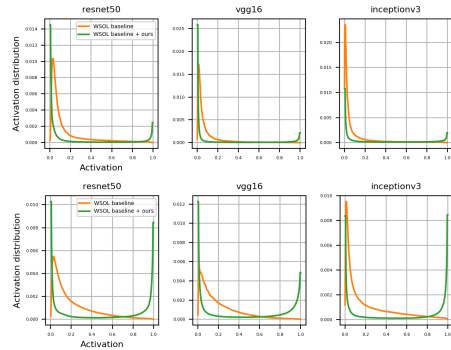

Figure 6: Distribution of activations over the test set for CAM* (orange) and CAM* + ours (green) for different backbones. Top: CUB. Bottom: OpenImages.

line compared to CAM* + ours[9]. For `MaxBoxAcc` metric, notice that the CAM*'s functioning region is concentrated near 0 while the localization performance completely drops to zero when τ starts to increase. This indicates that often the optimum threshold for these methods is very small. On the other hand, when combined with our method, `MaxBoxAcc` becomes less sensitive to τ. Over `PxAP` metric (Fig.5b), using our method improves the metric by pushing the curve to the top right corner. Following [7], we inspect the activation distribution of CAMs since they are related to the shape of the metrics curves (Fig.6). CAM*

shows a single mode with high concentration of activation near zero which explains the shape of the `MaxBoxAcc` and `PxAP` metrics (Fig.5). In addition, this makes the search of the optimum threshold difficult. Moreover, changing the threshold slightly could lead to high variation in localization performance since the threshold is expected to separate the foreground from background. The single mode in the distribution means that the method is unable to separate foreground and background regions, and reflects more uncertainty in CAM interpretability. On the other hand, using F-CAM allows the appearance of a second mode near one. (Mode near zero is for background pixels, while the mode near one is for foreground pixels.) This second mode is more clear over OpenImages dataset. Moreover, both modes are sharper than CAM* alone. This reflects assigning foreground and background to each pixel with more certainty. The reason for this behavior is that our activations are trained using cross-entropy (Eq.3) which pushes activations to be certain (0/1). These results explain why metrics over baseline CAM* perform well only near zero with large sensitivity to thresholding while our method shows less sensitivity. It is important to note that CAM activation distributions (Fig.6) are heavily influenced by the object size. For datasets with small objects, background pixels which are expected to have low activation magnitude will lead typically to density around zero that is higher than the density around one. This can be observed over CUB for our method. On the other hand, over datasets with large objects, density around one is typically expected to be high which is observed over OpenImages using our method.

Note: Supplementary materials contain failure cases, CRF loss, ablation studies, run-time analysis, results on the impact of CAM size vs. localization accuracy, convergence curves of WSOL baselines, and more visual results.

5. Conclusion

CAM methods typically rely on interpolation in order to restore full size CAMs in WSOL tasks. To improve the localization accuracy of CAMs, this paper proposes to connect a trainable decoding architecture to a CNN classifier, allowing for parametric upscaling of CAMs to accurate full resolution CAMs (F-CAMs). Low resolution CAMs and variants priors are used to fine-tune the decoder. Evaluated in combination with six baseline WSOL methods and three CNN backbones, our F-CAM methods improves the performance of these baselines by a large margin on CUB and OpenImage datasets. F-CAM performance is competitive with state-of-the-art WSOL methods, yet it requires less computation resources during inference.

---

[9]The rest of the methods are presented in the supplementary material.

References

[1] S. Belharbi, I. Ben Ayed, L. McCaffrey, and E. Granger. Deep ordinal classification with inequality constraints. *CoRR*, abs/1911.10720, 2019.

[2] S. Belharbi, I. Ben Ayed, L. McCaffrey, and E. Granger. Deep active learning for joint classification & segmentation with weak annotator. In *WACV*, 2021.

[3] S. Belharbi, J. Rony, J. Dolz, I. Ben Ayed, L. McCaffrey, and E. Granger. Deep interpretable classification and weakly-supervised segmentation of histology images via max-min uncertainty. *CoRR*, abs/2011.07221, 2020.

[4] R. Benenson, S. Popov, and V. Ferrari. Large-scale interactive object segmentation with human annotators. In *CVPR*, 2019.

[5] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[6] A. Chattopadhyay, A. Sarkar, P. Howlader, and V. N. Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *WACV*, 2018.

[7] J. Choe, S. J. Oh, S. Lee, S. Chun, Z. Akata, and H. Shim. Evaluating weakly supervised object localization methods right. In *CVPR*, 2020.

[8] J. Choe and H. Shim. Attention-based dropout layer for weakly supervised object localization. In *CVPR*, 2019.

[9] T. Deselaers, B. Alexe, and V. Ferrari. Weakly supervised localization and learning with generic knowledge. *IJCV*, 100(3):275–293, 2012.

[10] T. Durand, T. Mordan, N. Thome, and M. Cord. Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In *CVPR*, 2017.

[11] R. Fu, Q. Hu, X. Dong, Y. Guo, Y. Gao, and B. Li. Axiom-based grad-cam: Towards accurate visualization and explanation of cnns. *BMVC*, 2020.

[12] W. Gao, F. Wan, X. Pan, Z. Peng, Q. Tian, Z. Han, B. Zhou, and Q. Ye. TS-CAM: token semantic coupled attention map for weakly supervised object localization. *CoRR*, abs/2103.14862, 2021.

[13] K. He, X. Zhang, S.g Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[14] P.-T. Jiang, C.-B. Zhang, Q. Hou, M.-M. Cheng, and Y. Wei. Layercam: Exploring hierarchical class activation maps for localization. *IEEE Trans. Image Process.*, 30:5875–5888, 2021.

[15] H. Kervadec, J. Dolz, J. Yuan, C. Desrosiers, E. Granger, and I. Ben Ayed. Constrained deep networks: Lagrangian optimization via log-barrier extensions. *CoRR*, abs/1904.04205, 2019.

[16] M. Ki, Y. Uh, W. Lee, and H. Byun. In-sample contrastive learning and consistent attention for weakly supervised object localization. In *ACCV*, 2020.

[17] J. Lee, E. Kim, S. Lee, J. Lee, and S. Yoon. Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference. In *CVPR*, 2019.

[18] M. Lin, Q. Chen, and S. Yan. Network in network. *CoRR*, abs/1312.4400, 2013.

[19] J. Mai, M. Yang, and W. Luo. Erasing integrated learning: A simple yet effective approach for weakly supervised object localization. In *CVPR*, June 2020.

[20] A. Meethal, M. Pedersoli, S. Belharbi, and E. Granger. Convolutional STN for weakly supervised object localization. In *ICPR*, 2020.

[21] D. Omeiza, S. Speakman, C. Cintas, and K. Weldemariam. Smooth grad-cam++: An enhanced inference level visualization technique for deep convolutional neural network models. *CoRR*, abs/1908.01224, 2019.

[22] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[23] D. Pathak, P. Krahenbuhl, and T. Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *ICCV*, 2015.

[24] P. H. O. Pinheiro and R. Collobert. From image-level to pixel-level labeling with convolutional networks. In *CVPR*, 2015.

[25] A. Rahimi, A. Shaban, T. Ajanthan, R.I. Hartley, and B. Boots. Pairwise similarity knowledge transfer for weakly supervised object localization. In *ECCV*, 2020.

[26] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[27] J. Rony, S. Belharbi, J. Dolz, I. Ben Ayed, L. McCaffrey, and E. Granger. Deep weakly-supervised learning methods for classification and localization in histology images: a survey. *CoRR*, abs/1909.03354, 2019.

[28] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017.

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015.

[30] K.K. Singh and Y.J. Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *ICCV*, 2017.

[31] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958, 2014.

[32] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[33] T. Tagaris, M. Sdraka, and A. Stafylopatis. High-resolution class activation mapping. In *ICIP*, 2019.

[34] M. Tang, F. Perazzi, A. Djelouah, I. Ben Ayed, C. Schroers, and Y. Boykov. On regularized losses for weakly-supervised cnn segmentation. In *ECCV*, 2018.

[35] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021.

[36] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Deep image prior. *IJCV*, 128(7):1867–1888, 2020.

[37] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, California Institute of Technology, 2011.

[38] F. Wan, P. Wei, Z. Han, J. Jiao, and Q. Ye. Min-entropy latent model for weakly supervised object detection. *PAMI*, 41(10):2395–2409, 2019.

[39] J. Wei, Q. Wang, Z. Li, S. Wang, S. K. Zhou, and S. Cui. Shallow feature matters for weakly supervised object localization. In *CVPR*, 2021.

[40] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *CVPR*, 2017.

[41] Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng, and T. S. Huang. Revisiting dilated convolution: A simple approach for weakly- and semi-supervised semantic segmentation. In *CVPR*, 2018.

[42] H. Xue, C. Liu, F. Wan, J. Jiao, X. Ji, and Q. Ye. Danet: Divergent activation for weakly supervised object localization. In *ICCV*, 2019.

[43] S. Yang, Y. Kim, Y. Kim, and C. Kim. Combinational class activation maps for weakly supervised object localization. In *WACV*, 2020.

[44] S. Yang, Y. Kim, Y. Kim, and C. Kim. Combinational class activation maps for weakly supervised object localization. In *WACV*, 2020.

[45] F. Yu, V. Koltun, and T.A. Funkhouser. Dilated residual networks. In *CVPR*, 2017.

[46] S. Yun, D. Han, S. Chun, S.J. Oh, Y. Yoo, and J. Choe. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019.

[47] B. Zhang, J. Xiao, Y. Wei, M. Sun, and K. Huang. Reliability does matter: An end-to-end weakly supervised semantic segmentation approach. In *AAAI*, pages 12765–12772, 2020.

[48] C.-L Zhang, Y.-H. Cao, and J. Wu. Rethinking the route towards weakly supervised object localization. In *CVPR*, 2020.

[49] X. Zhang, Y. Wei, J. Feng, Y. Yang, and T.S. Huang. Adversarial complementary learning for weakly supervised object localization. In *CVPR*, 2018.

[50] X. Zhang, Y. Wei, G. Kang, Y. Yang, and T.S. Huang. Self-produced guidance for weakly-supervised object localization. In *ECCV*, 2018.

[51] X. Zhang, Y. Wei, and Y. Yang. Inter-image communication for weakly supervised localization. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *ECCV*, Lecture Notes in Computer Science, 2020.

[52] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016.

[53] Z.-H. Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2017.