MinMaxCAM: Improving object coverage for CAM-based Weakly Supervised Object Localization

Kaili Wang\textsuperscript{1,2}, Jose Oramas\textsuperscript{2}, Tinne Tuytelaars\textsuperscript{1}
\textsuperscript{1}KU Leuven, ESAT-PSI
\textsuperscript{2}University of Antwerp, imec-IDLab

Abstract

One of the most common problems of weakly supervised object localization is that of inaccurate object coverage. In the context of state-of-the-art methods based on Class Activation Mapping, this is caused either by localization maps which focus, exclusively, on the most discriminative region of the objects of interest, or by activations occurring in background regions. To address these two problems, we propose two representation regularization mechanisms: Full Region Regularization which tries to maximize the coverage of the localization map inside the object region, and Common Region Regularization which minimizes the activations occurring in background regions. We evaluate the two regularizations on the ImageNet, CUB-200-2011 and OpenImages-segmentation datasets, and show that the proposed regularizations tackle both problems, outperforming the state-of-the-art by a significant margin.

1 Introduction

Learning how to localize objects in images without relying on data paired with expensive location-specific annotations is a highly desirable capability. Therefore, it is no surprise that this task, usually referred to as Weakly-supervised Object Localization (WSOL), has gained attention in recent years.

One of the most used methods for this task is based on Class Activation Map (CAM) [Zhou et al., 2016], see [Singh and Lee, 2017; Zhang et al., 2018b; Zhang et al., 2018a; Choe and Shim, 2019; Yun et al., 2019; Zhang et al., 2020]. In these works, it has been noticed that the localization map generated by CAM focuses on the most discriminative region of the image. The reason is simple: the backbone is trained for classification since there is no access to the coordinates of the object, so it learns the discriminative features for each class. As a result, the object coverage is under-estimated.

Existing efforts to address this problem follow one of three common strategies. They either iteratively occlude/replace relevant regions of input in order to force the model to learn features that enable localization [Singh and Lee, 2017; Yun et al., 2019], or rely on additional networks to assist with the localization task [Zhang et al., 2018b; Zhang et al., 2018a]. Alternatively, a more simple strategy is to update the representation learned by the convolutional layers of the model [Choe and Shim, 2019; Zhang et al., 2020].

For CAM-based methods, the localization map is generated by a weighted linear combination of the feature map of the last convolutional layer and then rescaled to the image’s size. If the size of the feature map is too small, say $7 \times 7$ while the input size is $224 \times 224$, the resized localization map will have poor precision. In order to increase the size while still loading the original pre-trained weights of the backbone, a common strategy is to change the stride in some convolutional block [Choe and Shim, 2019; Zhang et al., 2020; Choe et al., 2020b]. However, we observe the localization map generated this way can activate a lot on the background, i.e. the object coverage is over-estimated (see Fig. 1). To solve it, activations on the background should be suppressed. To this end, [Yang et al., 2020] compute all possible CAM of one image and use some pre-defined combination functions to combine them. Clearly, this method costs more computation and the combination functions are not optimized.

Then the research question for us is to find a way to actively control the activation distribution on the localization map, maximizing or minimizing the activations as needed. We propose MinMaxCAM, a method that learns how to re-weight the feature maps initially learned by the classification model so that it is capable of shifting the mass of their internal activations. This not only enables accurate object localization but is relatively stable to train and does not need additional networks. In particular, we design two regularizations, Common Region Regularization (CRR) and Full Region Regularization (FRR), that can serve as objective functions for the model to optimize the linear layer after global average pooling (GAP). CRR encourages multiple images from the same class to share similar foreground representations: the representations extracted from the common foreground regions should be close to each other. During training, the generated localization map is used to extract the localized-object representation. By minimizing the distance between these representations, the model can optimize the localization map. On the other hand, inner-class differences can reduce the common region to a part of the object only. The same may happen due to failed localization of the most discriminative region. To tackle these situations, FRR is proposed. It stimulates covering a larger part of the object.
MinMaxCAM has a number of advantages: i) It is lightweight: it only relies on a standard classification model; no extra network is needed. It saves computation resources and is relatively simple to train. ii) The proposed method produces more precise or tighter bounding boxes. iii) Despite its simplicity, the proposed method is capable of setting a new state-of-the-art performance on the ImageNet, CUB-200-2011 and OpenImages-segmentation datasets, outperforming existing methods by a significant margin.

2 Related Work

Most existing works related to WSOL [Singh and Lee, 2017; Zhang et al., 2018a; Zhang et al., 2018b; Choe and Shim, 2019; Yun et al., 2019; Zhang et al., 2020; Yang et al., 2020] are based on Class Activation Mapping (CAM) [Zhou et al., 2016]. They address the WSOL task, indirectly, by solving the problem that the generated localization map only focuses on the most discriminative regions of the image. These methods can be divided into two types: non-parametric (w.r.t. CAM) and parametric methods. [Singh and Lee, 2017; Choe and Shim, 2019; Yun et al., 2019; Zhang et al., 2020] and our work belong to the first type. These methods do not need extra networks during the training and inference phases. This makes the methods lightweight, easy to implement and saves computation resources. [Singh and Lee, 2017] force the neural networks to focus on other relevant regions of the objects of interest by randomly occluding some patches of the input image when the network is trained for a classification task. [Yun et al., 2019] extend this idea by using patches from other images as occluding regions in a given image. [Choe and Shim, 2019] propose a simple but effective method: randomly drop out the most highly activated region or apply an attention mask on the feature maps when the classification network is trained. [Zhang et al., 2020] use information shared by two images from the same class to improve the localization map. They apply two constraints to improve the quality of the localization map. The first constraint is to learn the consistent features of two images of the same class by randomly sampling the features located in the most activated region and minimizing their distance. The second constraint is to compensate the limitation that features can only keep consistency within batches, where it learns a global class center for each class. [Yang et al., 2020] found that localization maps generated by CAM can also activate on the background. To suppress the activations of an image, they propose to compute all the possible CAMs firstly, and combine them via a combination function. This combination function is not learned during the training but pre-defined and related to the prediction probability of each possible class. Differently, our method only computes the localization map once for each image.

[Zhang et al., 2018a; Zhang et al., 2018b] add extra components based on the CAM model. [Zhang et al., 2018a] proposes a two-head architecture where the activation map generated by one stream is used to suppress the most discriminative region of the activation map generated by another one. By doing this, the model learns to use information from other relevant regions instead of the most discriminative one. [Zhang et al., 2018b] proposes to generate self-produced guidance (SPG) masks that separate the target object from the background. The masks are learned by the high-confidence regions within attention maps progressively and they also provide the pixel-level supervisory signal for the classification networks.

Different from these methods, we focus on the linear combination part of the CAM method rather than the feature extraction part, or the structure of the input images. The linear layer is optimized based on the proposed regularizations, which provides the optimal combination factors to generate the localization map. Similar to the non-parametric methods, our method does not have more trainable parameters. We only introduce two more hyper-parameters which are the weights for the two regularizations.

3 Methodology

3.1 Problem statement

Class Activation Map (CAM) [Zhou et al., 2016] is widely used to localize an object of interest in an image, in a weakly supervised manner. Given a backbone $B$ applied to input image $I$, followed by GAP and a linear layer $F$, in which $w \in \mathbb{R}^{C \times K}$ is the weight matrix, where $C$ is the number of classes. The CAM of an image is computed as:

$$CAM_{raw} = \sum_{k=0}^{K-1} w_c^k B(I)$$  \hspace{1cm} (1)

$$CAM = \frac{CAM_{raw} - \min(CAM_{raw})}{\max(CAM_{raw}) - \min(CAM_{raw})}$$  \hspace{1cm} (2)

where $c$ is the class of image $I$. In short, the localization map (CAM) is a linear combination of the feature maps of the last convolutional block of the backbone $B$. The weights of this combinations are taken directly from the weights of the linear layer w.r.t. the class which $I$ belongs to. [Choe et al., 2020b; Zhou et al., 2016] noted that localization maps generated by VGG often focus on the most discriminative region of the image, rather than on the whole object. We also observe that for different backbones $B$, the localization map can sometimes even cover the whole image, i.e. fail to localize the object. Fig. 1 shows some examples of the two cases. The second case has not been addressed in the literature yet. Here we propose two regularizations to address these problems.

![Figure 1: CAM localization maps generated with the VGG16 (top) and MobileNet (bottom) backbones with their estimated bounding boxes (red) and ground-truth (green).](image)


3.2 Common region regularization

The idea of the Common region regularization (CRR) is that different images depicting foreground objects of the same class should share similar features. The localized-object features can be obtained by applying the localization map $H$ which is generated via Eq. 1&2 on the image $I$ and extract the feature of $I \odot H$, denoted as $f$. Therefore, $f = B(I \odot H)$, where $\odot$ refers to the element-wise multiplication. To save computation resources, we use the same backbone $B$.

Given $S$ different images from the same class, we have

$$CRR = \frac{1}{S(S-1)} \sum_{i=0}^{S-1} \sum_{j=0}^{S-1} ||f_i - f_j||_2^2 \tag{3}$$

CRR calculates the pair-wise distance of the feature of $I \odot H$. The goal of CRR is to localize the common region of a set of images from the same class. By minimizing it, it can minimize the activations on the localization map, i.e. suppress the activations in non-object regions of the images.

There are two conditions in place for CRR to work: 1) a set of images from the same class should have different background and 2) $f$ should have different activation values for different backgrounds. The first assumption is dependent on the dataset. We will discuss the second assumption later.

3.3 Full region regularization

The goal of CRR is to make the localization map small in order to only focus on common regions of a set of images. However, this can have the side-effect of making the backbone only focus on a small part of the object if the objects have some inner difference in different images. In addition, for the case of the failed localization on the most discriminative region, we propose Full region regularization (FRR) to enlarge the localization map.

$$FRR = \frac{1}{S} \sum_{i=0}^{S-1} ||f_i - f_i^o||_2^2 \tag{4}$$

$f^o$ is the feature of the original image $I$, i.e. $f^o = B(I)$. FRR calculates the distance of the feature between the $I \odot H$ and $I$. Minimizing FRR has the effect of maximizing the activations on the localization map. Similar to CRR, there is one assumption in place for FRR to work: $B$ should not be invariant to changes in intensity. We will discuss it in Sec.4.4.

3.4 Learning process

The learning process has two stages. For stage I, it takes $N \times S$ images as input. The model is trained for the classification task, i.e. update the backbone and linear layer via the cross-entropy loss. It is the same as CAM.

$$L_{S1} = -\sum_{i=1}^{N \times S} c_i \log(y_i) \tag{5}$$

For stage II, the backbone $B$ is frozen as a feature extractor. It receives the images multiplied by the localization map $(I \odot H)$ as input. $H$ is generated via Eq. 1&2. This step introduces an intensity change on the original image. We discuss its effects in Sec. 4.4. The features ($f$ and $f^o$) extracted by $B$ are used for the two regularizations. The loss only updates the weights of the linear layer, by minimizing CRR and FRR.

$$L_{S2} = \lambda_1 CRR + \lambda_2 FRR \tag{6}$$

$L_{S1}$ and $L_{S2}$ update the model every mini-batch. Fig. 2 shows the proposed method. To train our model there is no extra hyper-parameters besides the weights for the two regularizations. During testing, the localization map is generated via CAM (Eq. 1&2), therefore, there is no need for a set of images per class.

**Why freeze the backbone ($B$) in Stage II?**

The goal of stage II is to adjust the weights $w^o_k$ which are used to generate the localization map by minimizing CRR and FRR. $B$ serves as a feature extractor in this stage. If $B$ were also updated, after minimizing FRR, it would become invariant to intensity changes. In the later training process, FRR could then not measure the difference between $f^o$ and $f$.

4 Experiments

4.1 Dataset and performance metric

We consider three widely used datasets ImageNet [Deng et al., 2009], CUB-200-2011 (CUB) [Wah et al., 2011] and OpenImages instance segmentation subset [Benenson et al., 2019; Choe et al., 2020b] to evaluate our method. ImageNet contains 1,000 classes with over 1 million images. Following [Choe et al., 2020b], we use ImageNetV2 [Recht et al., 2019] as the validation set to tune our model. This validation set contains 10 images per class with the object bounding boxes annotated by I [Choe et al., 2020b]. CUB has 200 fine-grained classes of birds with 5,994 images for training and 5,794 images for testing. Similarly, we follow [Choe et al., 2020b] to use a validation set collected by them to tune the model. The validation set contains 1,000 images in total, around 5 images per class. OpenImages instance segmentation subset (OpenImages) I [Choe et al., 2020a] covers 100 classes. It contains 29,819, 2,500 and 5,000 images for training, validation and testing, respectively. Every image has the object segmentation as annotation.
Performance metric

[Zhou et al., 2016; Choe and Shim, 2019; Yun et al., 2019; Zhang et al., 2018a; Singh and Lee, 2017; Zhang et al., 2020] use a pre-defined threshold (0.2) for the generated CAM to produce a localization region. [Choe et al., 2020b] argues that using a fixed pre-defined threshold can be disadvantageous for certain methods since the ideal threshold may depend on the data and architecture that are used. In short, for different datasets, architectures and methods, the ideal threshold is different. We follow this idea and use the metric proposed by [Choe et al., 2020b]. For ImageNet and CUB datasets we use two threshold-free metrics to evaluate the localization map, i.e. MaxBoxAcc and MaxBoxAccV2. MaxBoxAcc is equivalent to GT-known localization accuracy where one localization map is counted correct when the intersection over union (IoU) of the estimation and ground truth bounding box is larger than 0.5. Differently, to avoid using a fixed pre-defined threshold for binarizing the localization map, here we set various $\tau$ thresholds to find the best performance. MaxBoxAccV2 is the average of three MaxBoxAcc when the IoU is 0.3, 0.5 and 0.7. For OpenImages dataset, since we have access to the segmentation mask, we use the pixel averaged precision ($PxAP$) proposed by [Choe et al., 2020b]. Similarly, $PxAP$ is also threshold-free. Please refer to [Choe et al., 2020b] for more details.

4.2 Implementation Details

We consider three different backbones: VGG16 [Simonyan and Zisserman, 2014], ResNet50 [He et al., 2015] and the lightweight MobilenetV2 [Sandler et al., 2018]. Following [Choe et al., 2020b; Zhang et al., 2020], for ResNet50 and MobilenetV2 we increase the size of the last feature map by changing the stride of convolution layers from 2 to 1. By doing this, the model can still load the pretrained weights. We set set size $S=5$, $N=12$ (i.e. batch size $=60$) for all the experiments except those with the ResNet50 backbone, where $N=10$ due to GPU memory limitations. For $\tau$ we set 100 intervals between 0 and 1.

4.3 Comparison with State-of-the-art Methods

In this part, we compare the proposed method w.r.t. several state-of-the-art methods on ImageNet, CUB and OpenImages datasets. Table 1 shows quantitative results. The quantitative results from these methods except I2C are taken from [Choe et al., 2020b; Choe et al., 2020a] where the authors used the validation set to select the final models. We implement the I2C method and use their suggested hyper-parameters to train the models. The results clearly show that our method outperforms the competing methods on all three datasets except when ResNet50 is selected as backbone for the CUB dataset. In this case, our method is only lower by 0.1 pp, in MaxBoxAcc w.r.t. Ha$\$S. Interestingly, for ImageNet with a ResNet50 backbone, only our method outperforms CAM. We believe it is due to the proposed CRR which minimizes the activations in the background. It is expected that I2C works better when MobilenetV2 or ResNet50 are used as backbone, since the constraints proposed by I2C prevent the model from activating highly in background regions, which is a weakness that Mobilenet and ResNet suffer from. Please note the performance can be further improved when the optimal hyperparameters are found.

Fig. 3 shows qualitative comparisons w.r.t. CAM on the ImageNet, CUB and OpenImages datasets. It clearly shows that our method can enlarge $H$ when it originally focuses on a small region and reduces it when it is highly-activated in the background. In the fifth-column example from the ImageNet dataset generated by the VGG backbone, the effect of different optimal thresholds $\tau$ can be noticed. The same effect can be seen in the Mobilenet-based localization map of the last example in the ImageNet dataset. In addition, in some cases although the estimated bounding box of CAM has a large IoU with the ground truth, the object region has a stronger activation for our method (e.g. the last example of the ImageNet dataset).

4.4 Analysis of our method

In this section we analyze several aspects of our method. The experiments are conducted on the CUB dataset.

Intensity change

For FRR to work, $B$ should be sensitive to changes in intensity of its input. To verify this assumption, we conduct an experiment where we add an intensity change as one of the data augmentations in stage I. In stage II we do not change anything. By doing this, we force $B$ to become more robust to intensity changes on its input (Please note, we cannot make $B$ completely robust). We use VGG16 as backbone. The performance drops 2.5pp and 1.9pp for MaxBoxAcc and MaxBoxAccV2, respectively. This shows that indeed the performance...
drops when $B$ becomes more robust to intensity changes.

**Background activation**

We have introduced in Sec. 3.2 that $B$ should activate differently for different background regions for CRR to work. If the assumption holds, then $f \in \mathbb{R}^K (f = B(I; C))$ to work. The higher the proportion, the more background is covered in the predicted bounding box that cover background regions. Please refer to the supplementary material for more details.

**Masking inputs vs. masking features**

In order to get the localized-object representation $f$, we suggest applying element-wise product between $H$ and the input image $I$ firstly and then send it to the same backbone $B$. It can be argued that $H$ can be applied on the extracted feature map (noted as $f' \in \mathbb{R}^{N \times C \times H \times W}$) from stage I directly, which can avoid the re-computation of the feature. However, this may produce some problems.

On the one hand, $f'$ not only has spatial information but also has $C$ channels (around 1024 or 2048 for our backbones), and different channels can represent different concepts of the image [Oramas M et al., 2019]. On the other hand, the localization map $H \in \mathbb{R}^{H \times W}$ only contains spatial information, no channel-wise information. Information can be lost if $H$ is directly applied on $f'$ since the activations on the same spatial location but different channels will be encouraged/suppressed in the same scale. To verify our analysis, we conduct an experiment where we apply $H$ directly on $f'$. Table 2 suggests that our analysis is correct. Especially, for VGG16 the performance is even worse than CAM.

### 4.5 Ablation study

**Effects of CRR and FRR for different backbones ($B$)**

We analyze the effect of CRR and FRR on different backbones. In Sec. 3.1 we mentioned the issue of the localization map $H$ generated by different architectures with Fig. 1 showing some examples. To verify that the failure cases are consistent across the whole dataset, we analyse how large is the background covered by the estimated bounding boxes inferred from the VGG16 and MobilenetV2 backbones.

For each test image, we compute the proportion of pixels in the predicted bounding box that cover background regions. The higher the proportion, the more background is covered by the estimated bounding box.

For MobilenetV2, roughly half of the images have a proportion above 0.1, while that happens only for roughly a quarter of the images for VGG16. This proves that indeed the localization map generated by MobilenetV2 activates more in the background region. Please refer to the supplementary material for more details.
We fix one regularization and ablate the other. and MobilenetV2 (bottom) backbones, respectively. For each plot, CRR

Figure 5: Ablation study w.r.t. CRR and FRR for the VGG16 (top) and MobilenetV2 (bottom) backbones, respectively. For each plot, We fix one regularization and ablate the other.

In practice, intuitively, if the localization map $H$ always localizes the most discriminative region of the object, FRR should play a more important role in the training process. On the contrary, $CRR$ should be more essential if $H$ is relatively highly activated in background regions. To verify the influence of $CRR$ and $FRR$, we gradually increase/decrease the weight of one regularization with the other one fixed.

Fig. 5 shows the performance curve. For VGG16, performance decreases gradually as the weight for $CRR$ increases. The opposite occurs with $FRR$. The performance increases and reach the peak when the weight of $FRR$ is 10, afterwards the performance decreases slightly (weight=20). On the contrary, increasing the weight of $CRR$ boosts the performance for MobilenetV2. The performance decreases slightly when the weight of $FRR$ is 0 while it drops dramatically when a larger weight is applied. These trends are expected since VGG16 focuses on small discriminative regions rather than the whole object while MobilenetV2 activates frequently on the background. In addition, the experiment with MobilenetV2 shows that $FRR$ can compensate the performance when a large $CRR$ is needed and applied.

Effect of the set size ($S$)

Here we discuss how the set size $S$ influences the performance of $CRR$. $CRR$ can benefit from using a relatively large $S$ because more representations can be grouped together simultaneously. We conduct series of experiments, where we decrease $S$ from 5 to 2 gradually. We use Mobilenet as backbone and keep the batch size the same (60 images). The results in Table 3 indicate that for a larger set size $S$, $FRR$ indeed works better.

| Method          | MaxBoxAcc (%) | MaxBoxAccV2 (%) |
|-----------------|---------------|-----------------|
| VGG-Ours        | 83.9          | 70.2            |
| Apply $H$ on $f'$ | -19.1         | -12.5           |
| MobilenetV2-Ours | 75.8          | 65.0            |
| Apply $H$ on $f'$ | -4.1          | -3.0            |

Table 2: Masking inputs vs. masking features

| Set size $S$ | MaxBoxAcc (%) | MaxBoxAccV2 (%) |
|--------------|---------------|-----------------|
| $S=5$        | 75.8          | 65.0            |
| $S=4$        | 74.7          | 64.6            |
| $S=3$        | 74.4          | 64.4            |
| $S=2$        | 73.9          | 63.9            |
| CAM          | 65.3          | 58.1            |

Table 3: Effect of the set size $S$.

| Method          | Top 1 Loc. / Top 5 Loc. |
|-----------------|-------------------------|
| VGG-AoL         | 45.9 / -                |
| VGG-ADL         | 52.3 / -                |
| VGG-CCAM        | 50.1 / 63.8             |
| VGG-Ours        | 66.0 / 83.9             |
| MobilenetV1-HaS | 44.7 / -                |
| MobilenetV1-ADL | 47.7 / -                |
| MobilenetV1-Ours| 54.8 / 69.4             |

Table 4: Top-1 and Top-5 localization rate.

Figure 6: Failure cases of the proposed method.

4.6 Classification v.s. localization

The classification task sometimes rely on information from the background [Oramas M et al., 2019], while the localization task only focuses on the foreground object. Therefore, a good localization model is not necessarily a good classification model. The evaluation metric top-1/5 Loc. takes into account both localization and classification accuracy of a given localization model, therefore, it is not able to accurately measure the localization performance [Choe et al., 2020b]. Top-1/5 Loc. can be low because of the classification accuracy even if the localization accuracy is good.

In order to compute Top-1/5 Loc. fairly, we propose a simple path: train a separate classification model to provide the predicted class for the object localization model. In practice, we use ResNet50 to train the classifier, whose classification accuracy on CUB dataset is 77.3%. Table 4 shows the Top-1/5 Loc. results. In order to compare with the competitive methods, here we use MobilenetV1 as backbone. The numbers of the competitive methods are taken from [Choe and Shim, 2019; Yang et al., 2020].

4.7 Failure cases

We analyze some failure cases in this section. The first type is caused by nature, which is unavoidable, like reflections of water. The cause of the second type is related to our first assumption for $CRR$, that different images from a class have very similar background. For example, brown creeper always appear with trunks in the image. Fig. 6 shows some examples.

5 Conclusion

We propose two representation regularizations, Common Region Regularization and Full Region Regularization, to overcome the weaknesses of weakly supervised object localization methods based on CAM. Our method relies only on a standard classification model; no extra network is needed. Through extensive experiments and analysis, we discuss relevant aspects of our method and show that it is capable of surpassing the state-of-the-art by a significant margin.
References

[Benenson et al., 2019] Rodrigo Benenson, Stefan Popov, and Vittorio Ferrari. Large-scale interactive object segmentation with human annotators. CVPR, 2019.

[Choe and Shim, 2019] Junsuk Choe and Hyunjung Shim. Attention-based dropout layer for weakly supervised object localization. 2019.

[Choe et al., 2020a] Junsuk Choe, Seong Joon Oh, Sanghyuk Chun, Zeynep Akata, and Hyunjung Shim. Evaluation for weakly supervised object localization: Protocol, metrics, and datasets. arXiv preprint arXiv:2007.04178, 2020.

[Choe et al., 2020b] Junsuk Choe, Seong Joon Oh, Seungho Lee, Sanghyuk Chun, Zeynep Akata, and Hyunjung Shim. Evaluating weakly supervised object localization methods right. In CVPR, 2020.

[Deng et al., 2009] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In CVPR09, 2009.

[He et al., 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. CoRR, 2015.

[Oramas M et al., 2019] José Oramas M, Kaili Wang, and Tinne Tuytelaars. Visual explanation by interpretation: Improving visual feedback capabilities of deep neural networks. In ICLR, 2019.

[Recht et al., 2019] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? ICML, 2019.

[Sandler et al., 2018] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. CVPR, 2018.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556, 2014.

[Singh and Lee, 2017] 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, 2017.

[van der Maaten and Hinton, 2008] L.J.P. van der Maaten and G.E. Hinton. Visualizing high-dimensional data using t-sne. Journal of Machine Learning Research, 2008.

[Wah et al., 2011] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011.

[Yang et al., 2020] Seunghan Yang, Younghyung Kim, Youngun Kim, and Changick Kim. Combinational class activation maps for weakly supervised object localization. WACV, 2020.

[Yun et al., 2019] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In ICCV, 2019.

[Zhang et al., 2018a] Xiaolin Zhang, Yunchao Wei, Jiashi Feng, Yi Yang, and Thomas Huang. Adversarial complementary learning for weakly supervised object localization. In CVPR, 2018.

[Zhang et al., 2018b] Xiaolin Zhang, Yunchao Wei, Guoliang Kang, Yi Yang, and Thomas Huang. Self-produced guidance for weakly-supervised object localization. In ECCV, 2018.

[Zhang et al., 2020] Xiaolin Zhang, Yunchao Wei, and Yi Yang. Inter-image communication for weakly supervised localization. In ECCV. Springer, 2020.

[Zhou et al., 2016] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In CVPR, 2016.