Adversarial Localization Network

Lijie Fan
Tsinghua University
flj14@mails.tsinghua.edu.cn

Shengjia Zhao
Stanford University
sjzhao@stanford.edu

Stefano Ermon
Stanford University
ermon@stanford.edu

Abstract

We propose the *Adversarial Localization Network*, a novel weakly supervised approach to generate object masks in an image. We train a corruption net to mask out regions of the image to reduce prediction confidence of the classifier. To avoid generating adversarial artifacts due to vulnerability of deep networks to adversarial perturbation, we also co-train the classifier to correctly predict the corrupted images. Our approach could efficiently produce reasonable object masks with a simple architecture and few hyper-parameters. We achieve competitive results on the ILSVRC2012 dataset for object localization with only image level labels and no bounding boxes for training.

1 Introduction

Supervised convolutional neural networks have achieved near human performance on several computer vision tasks, such as object localization [12, 15, 9, 26, 14] and semantic segmentation [16]. However, localization and segmentation annotations are often expensive and hard to obtain (compared to image-level labels such as object classes). To resolve this problem, weakly-supervised approaches are developed using image level labels only. These weakly supervised methods have reached competitive performance on various computer vision tasks, such as object detection [17, 27, 21] and semantic segmentation [18, 13].

A traditional approach [2, 3, 5] to weakly supervised localization or segmentation is to perturb (e.g., by occlusion) regions of the image to maximally decreases a classifier's prediction confidence. The assumption is that regions containing the object are the most important one for the decision, and as a result, they will significantly decrease a classifier's prediction confidence if perturbed. However, it has been shown that classifiers are very easily "fooled" [4] as a very small or even imperceptible amount of adversarial perturbation can completely alter a classifier's prediction. This effect, which we shall refer to as vulnerability to adversarial noise, is attributed to the fact that in high dimensions, there likely exists a direction where a small input change leads to a significant change in the output [6]. Therefore previous methods partition the space into very coarse grids, and apply the same perturbation to *all* pixels in each grid. This limits the dimensionality of the adversarial noise and alleviates this problem. However this usually means very coarse grained and inaccurate localizations or segmentations.

This paper presents two contributions that improve on the shortcomings of previous models. We draw inspiration from Virtual Adversarial Training (VAT)[8], which uses adversarial training to make a classifier more robust against adversarial noise. We perform adversarial training [24] between corruption network and classifier, so that a classifier's decisions will only be altered if the image is truly corrupted beyond recognition, which in our experiments, often leads to occlusion of the correct object. This allows us to learn much finer grained object masks compared to previous work [2, 3, 5]. In addition, we use super-pixels which incorporate structure about object boundary, rather than equally sized grids. The resulting mask follows the object contour better given the same resolution. In our experiments on the ILSVRC 2012 dataset, our method obtains superior or comparable results

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

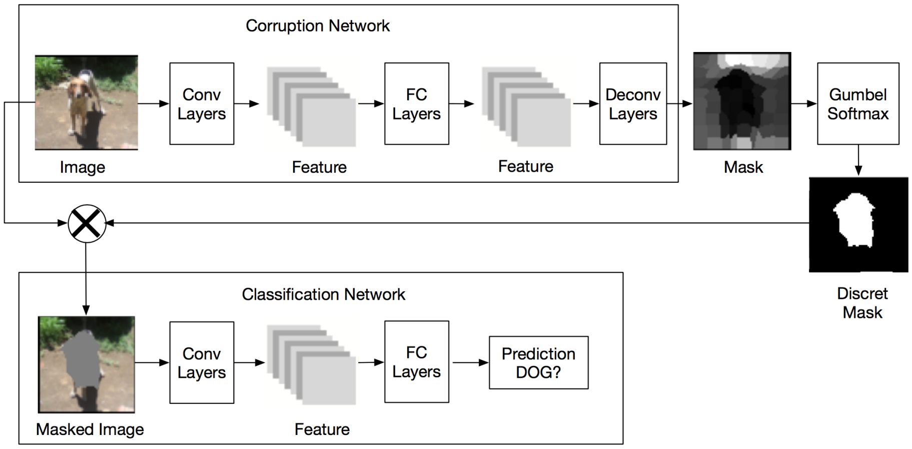

Figure 1: Adversarial Localization Network: The overall architecture of our proposed approach. The corruption network takes original images as inputs and produces corresponding saliency masks. The classification takes the masked images as input and produces classification scores. These two networks are trained in an adversarial way.

with previous state-of-the-art with a fairly small model architecture with few hyper-parameters or post processing.

2 Proposed Approach

In this section, we give detailed descriptions of our approach and design choices. Our model is shown in Figure 1. We train two components, a "corruption process" m_ϕ and a "classification network" c_θ. The corruption process consistent of three parts.

1) A deep network parameterized by ϕ takes the image x as input, and outputs a probability of corruption for each pixel. Details about the corruption network are given in Section 2.1.

2) Segment the image into super-pixels. For each super-pixel, we merge (by averaging) the occlusion probability of all its included pixels, and sample a binary decision to occlude or not occlude based on the occlusion probability. This procedure is described in Section 2.2.

3) We apply the occlusion mask to the image by setting occluded pixels to value 0. We denote the resulting corrupted image as $\tilde{x} = m_\phi(x)$.

On the other hand, the classifier network c_θ takes as input the corrupted image $\tilde{x}$, and outputs the resulting logits scores $c_\theta(\tilde{x})$ where $c_\theta(\tilde{x})_j$ is the classifier's prediction for the j-th class. Let y be the ground truth label. The classifier net is trained to increase the softmax cross entropy with the true label $\mathrm{Softmax\_CE}(c_\theta(\tilde{x}), y)$, while the corruption network is trained to reduce prediction confidence. The training procedure and regularizations used are described in Section 2.3.

2.1 Corruption Network

Because the extent of an object depends both on local information (such as position of edges) and global informations (such as object category), we design the corruption network to capture both global and local information. To capture global information, we first use a convolutional network with fully connected layers to map the input into high level features, then we use the inverted architecture (deconvolution) to decompress high level features into occlusion probabilities for each pixel. This process with fully connected layers ensures that we are able to model global information about objects (such as its category). However this also implies that details such as exact edge positions will get lost in the convolution-pooling process. Empirically we observe that such architecture produces inaccurate object boundaries. Therefore we also add a large number of residual connections in the intermediate layers. This ensures that local information can be more easily forwarded and utilized.

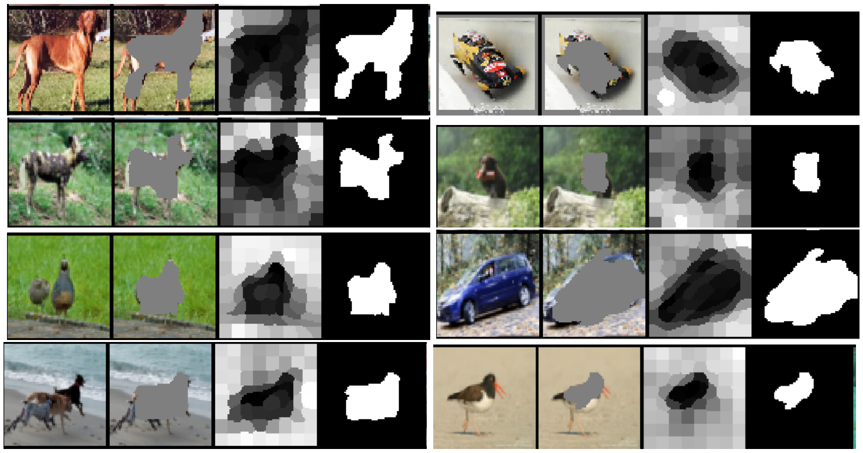

Figure 2: Examples of the generated results of our Adversarial Localization Network: First column: original input images; Second column: the masked images; Third column: mask intensity maps; Fourth column: the generated object segmentation mask.

2.2 Pixel Level Probabilities to Masks

We design a procedure to convert pixel level probabilities produced by the corruption network into (discrete) masks that are consistent with the input image, and allow for efficient training with back-propagation.

One observation is that object boundaries almost always align with image edges (points with sharp transitions of color or intensity). Therefore the edges of the image naturally partition the image into super-pixels(group of pixels). We use the super-pixel computation method proposed in [11], and each super-pixel can be treated as a whole when deciding whether it belongs to the object. Therefore we propose to average the masking probabilities for each super-pixel, and produce a masking decision for each super-pixel as a whole. This reduces the dimensionality of possible perturbations which alleviates the vulnerability to adversarial perturbations.

Next, we discretize the masking probability for each super-pixel to a binary masking decision (mask or not mask) using the Gumbel trick [10] with the straight-through estimator [7]. This is described by the following procedure. Let $\boldsymbol{p} = (p, 1-p)$ be a probability vector of a Bernoulli variable. Sample.

$$u_1, u_2 \sim \text{Uniform}\,(0, 1)\,, \qquad z_i = -ln\,(-ln\,(-u_i))\,, i = 1, 2$$

Then we draw a sample x from the Bernoulli distribution $\boldsymbol{p}$ by

$$x = \mathbb{I}(p + z_1 > 1 - p + z_2)$$

where $\mathbb{I}$ is the indicator variable. Gradient through this discretization indicator operation can be approximated by the straight-through estimator [7].

2.3 Adversarial Training Procedure

As described above, we train the networks with an adversarial procedure. For the corruption network, for each data point, label pair x, y we minimize

$$\min_{\phi} \mathcal{L}_{\text{Corruption}} = \sum_{j=1}^{K} \mathbb{I}_j c_\theta(m_\phi(x))_j - \frac{1}{K-1}(1 - \mathbb{I}_j) c_\theta(m_\phi(x)) \tag{1}$$

where K is the number of object classes, and $\mathbb{I}_j$ is the indicator variable of whether class j is the top prediction made by the network: $\mathbb{I}_j = 1$ if $j = \arg\max_{1 \leq i \leq K} c_\theta(m_\phi(x))_j$. Intuitively this objective encourages the corruption network to corrupt the image such that the classifier is in a maximum state of confusion: it produces the same logit score for each class. For the classifier network, we simply train to maximize the softmax cross entropy with the true label

$$\max_{\theta} \mathcal{L}_{\text{classifier}} = \text{Softmax\_CE}(c_\theta(m_\phi(x)), y)$$

One trivial solution is for the mask to occlude the entire image, so that classifier has no information at all to make any prediction, minimizing the corruption network loss. Therefore add a regularization term that penalizes the total area of the generated masks so that the corruption network will only generate the minimum mask necessary to confuse the classifier.

$$\min_{\phi} \mathcal{L}_{\text{Corruption}} + \lambda_{\text{Reg}} \sum_{i}^{\text{all pixels}} \mathbb{I}(i \text{ is occluded})$$

where $\mathbb{I}$ is the indicator variable of events. This results in well-interpretable image masks that often occludes the correct object.

2.4 Balancing Adversarial Training

A practical concern of our method, is convergence to undesired local optimum. For example, the corruption net may occlude nothing, which is a local optima because adding a small amount of occlusion will lead to very little increase in classifier confusion, but incurs a penalty because we penalize the area of the occlusion mask. On the other hand, the corruption net may occlude too much if the penalty weighting λ_{Reg} is too small. We implemented an adaptive strategy to fix this weighting. We observe that the best results are usually obtained when the classifier accuracy on the corrupted image is around 10%, so we decrease (so occlude more) or increase (so occlude less) the weighting λ_{Reg} depending on the classifier accuracy.

Another concern of adversarial training is the delicate equilibrium that is often broken if either component trains significantly faster. In our implementation we observe that the classifier usually trains significantly faster than the corruption network. This causes instability and forces the occlusion network to generate a larger mask. We found the above adaptive strategy to work well for this issue as well: if the classifier accuracy is above 10% we only train the corruption net. If accuracy is below 10% we perform joint training. We have found that this simple strategy significantly improves training stability and quality of generated masks.

3 Experiments

We conduct our experiments on ILSVRC 2012 localization dataset [19] with 1000 classes and 1.2 million images. We train our model on the training set with only image level class labels, and evaluate on localization labels provided in the validation set.

3.1 Evaluation metric

We use the following two types of evaluation metrics:

Metric 1: Top-1 accuracy This is the localization evaluation metric officially provided by ILSVRC. For each test image, provide a classification prediction and one object bounding box prediction. Compute the percentage of test samples which are classified correctly and have more than 50% intersection over union (IoU) between the predicted bounding box and one of the ground-truth bounding boxes.

Metric 2: Top-1 accuracy with state-of-the-art (SOTA) classifier We train our network with a fairly simple classifier for stability and speed of the adversarial training procedure. Therefore performance can be improved when we replace the class prediction made by our smaller network with that made by a state-of-the-art classifier, while keeping the same bounding box prediction. We use a InceptionV4 network [22] to produce the class prediction instead.

3.2 Bounding Box Generation

We convert masking probabilities to a bounding boxes in our localization experiment. To do this we first binarize the masking probability map with an adaptive threshold.

$$\text{Threshold} = 0.7 \times \text{Mean}(\text{map}) + 0.3 \times \text{Max}(\text{map})$$

Then we take the bounding box that covers the largest connected component as our prediction.

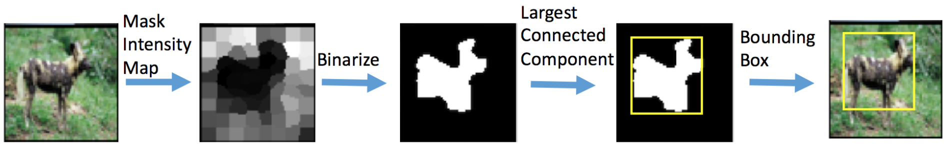

Figure 3: Bounding box generation pipeline: 1.Generate masking probability map with the corruption network; 2.Binarize the map with adaptive thresholds; 3.Compute the Largest Connected Component in the binarized mask; 4.Generate its bounding box.

3.3 Comparison with the baseline model

We compare with other state-of-the-art weakly supervised localization models [20, 23, 27]. The results are shown in Table 1. Our Adversarial Localization Network performs 3.1% and 1.9% better than the baseline model on Top-1 accuracy with SOTA classifier and Top-1 accuracy respectively. Importantly. Note that we train on a much smaller architecture compared to baseline models: the input is a 64×64 image and we use a simple 4-layer convolutional network. Each evaluation of our model only takes 0.093 seconds while the fastest baseline model takes 0.14 seconds. This indicates significant speed up with similar or even better performance.

Methods	Metric 1	Metric 2
Max box	41.0%	34.3%
Backprop [20]	–	38.7%
Layer-wise Relevance Propagation [23]	42.2%	–
Global Average Pooling(Baseline)[27]	51.9%	43.6%
Adversarial Localization Networks	**56.5%**	**45.5%**

Table 1: Performance comparison between Adversarial Localization Network(ALN) and state-of-the-art models

3.4 Necessity of Adversarial Training

To demonstrate the necessity of our adversarial training procedure, we only train the corruption network with the same loss as Eq.(1). We show that it is very easy for the corruption net to generate adversarial artifacts to "fool" the classification network, even when we are using super-pixel representation for the input image, or use increased penalty for the size of the generated mask. If the classification network is also trained, almost no adversarial artifact are observed.

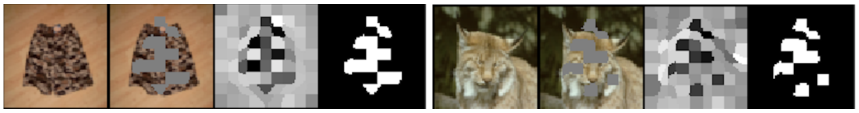

Figure 4: Adversarial Examples: To show the necessity of adversarial learning, if we freeze the classification network and only train the corruption network, it could be trained to generate adversarial examples to "fool" the classifier easily, however, those adversarial examples make no sense to humans.

4 Conclusion

In this work, we presented a novel weakly supervised approach for object localization/segmentation. In particular, we make two major contributions: applying adversarial training to avoid vulnerability of deep networks to adversarial perturbation and make classifier more robust, and using super-pixels, which reduces the dimensionality of possible perturbations, alleviates the vulnerability to adversarial perturbations and leads to better detection boundary. We out-perform state-of-the-art models in performance and have significantly faster training and evaluation time.

References

[1] M. Van den Bergh, and X. Boix, and G. Roig and B. de Capitani and L. Van Gool, “SEEDS: Superpixels extracted via energy-driven sampling,” *European conference on computer vision*, 2012

[2] P. Dabkowski, and Y. Gal, “Real Time Image Saliency for Black Box Classifier,” *arXiv preprint arXiv:1705.07857*, 2017.

[3] R. Fong, and A. Vedaldi , “Interpretable Explanations of Black Boxes by Meaningful Perturbation,” *arXiv preprint arXiv:1704.03296*, 2017.

[4] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, “Universal adversarial perturbations,” *arXiv preprint arXiv:1610.08401*, 2016.

[5] 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,” *arXiv preprint arXiv:1703.08448*, 2017.

[6] I. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” *arXiv preprint arXiv:1412.6572*, 2016.

[7] Y. Bengio, N. Léonard, and A. Courville, “Estimating or propagating gradients through stochastic neurons for conditional computation,” *arXiv preprint arXiv:1308.3432*, 2013.

[8] T. Miyato, S.-i. Maeda, M. Koyama, K. Nakae, and S. Ishii, “Distributional smoothing with virtual adversarial training,” *arXiv preprint arXiv:1507.00677*, 2015.

[9] K. Duan, D. Parikh, D. Crandall, and K. Grauman. Discovering localized attributes for fine-grained recognition. In *CVPR*, 2012.

[10] E. J. Gumbel, *Statistical theory of extreme values and some practical applications: a series of lectures*. US Govt. Print. Office, 1954, no. 33.

[11] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “Slic superpixels compared to state-of-the-art superpixel methods,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[12] R. Girshick. Fast r-cnn. In *ICCV*, 2015.

[13] A. Khoreva, R. Benenson, M. Omran, M. Hein, and B. Schiele. Weakly supervised object boundaries. In *CVPR*, 2016.

[14] M. Kiapour, K. Yamaguchi, A. C. Berg, and T. L. Berg. Hipster wars: Discovering elements of fashion styles. In *ECCV*, 2014.

[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016.

[16] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[17] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Is object localization for free? – weakly-supervised learning with convolutional neural networks. In *CVPR*, 2015.

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

[19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.

[20] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR Workshop*, 2014.

[21] K. K. Singh, F. Xiao, and Y. J. Lee. Track and transfer: Watching videos to simulate strong human supervision for weakly-supervised object detection. In *CVPR*, 2016.

[22] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning.” in *AAAI*, 2017, pp. 4278–4284.

[23] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, “On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation,” *PloS one*, vol. 10, no. 7, p. e0130140, 2015.

[24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[25] P. Pinheiro and R. Collobert, “Recurrent convolutional neural networks for scene labeling,” in *International Conference on Machine Learning*, 2014, pp. 82–90.

[26] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev. PANDA: Pose Aligned Networks for Deep Attribute Modeling. In *CVPR*, 2014.

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