Generative Adversarial Learning Towards Fast Weakly Supervised Detection

Yunhan Shen[1] Rongrong Ji[1*] Shengchuan Zhang[1] Wangmeng Zuo[2], Yan Wang[3]
[1]Fujian Key Laboratory of Sensing and Computing for Smart City,
School of Information Science and Engineering, Xiamen University, Fujian, China
[2]School of Computer Science and Technology, Harbin Institute of Technology, China
[3]Microsoft, Redmond, USA
yhshen@stu.xmu.edu.cn rrji@xmu.edu.cn zsc_2007@163.com wmzuo@hit.edu.cn wanyan@microsoft.com

Abstract

Weakly supervised object detection has attracted extensive research efforts in recent years. Without the need of annotating bounding boxes, the existing methods usually follow a two/multi-stage pipeline with an online compulsive stage to extract object proposals, which is an order of magnitude slower than fast fully supervised object detectors such as SSD [31] and YOLO [34]. In this paper, we speedup online weakly supervised object detectors by orders of magnitude by proposing a novel generative adversarial learning paradigm. In the proposed paradigm, the generator is a one-stage object detector to generate bounding boxes from images. To guide the learning of object-level generator, a surrogator is introduced to mine high-quality bounding boxes for training. We further adapt a structural similarity loss in combination with an adversarial loss into the training objective, which solves the challenge that the bounding boxes produced by the surrogator may not well capture their ground truth. Our one-stage detector outperforms all existing schemes in terms of detection accuracy, running at 118 *frames per second, which is up to* 438× *faster than the state-of-the-art weakly supervised detectors [8, 30, 15, 27, 45]. The code will be available publicly soon.*

1. Introduction

Weakly supervised object detection (WSD) has attracted extensive attention in the recent years [8, 28, 45, 10, 49, 27]. A significant advantage of WSD lies in removing the necessity of labor-intensive annotation of object bounding boxes. Instead, it exploits image-level annotations that are widely available from the Internet. To take advantage of such image-level weak supervision, most previous methods [49, 8, 45] use a Multiple Instance Learning (MIL) pipeline.

*Corresponding author.

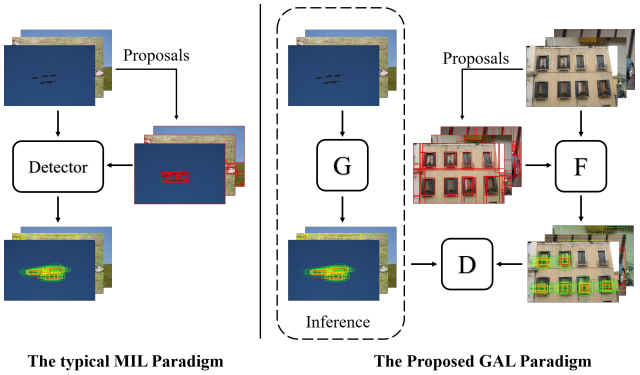

Figure 1: Comparison between the classic weakly supervised detectors and our method. The typical frameworks employ a Multiple-Instance Learning paradigm, and are two/multi-stage detectors during training and testing. Our method adopts a Generative Adversarial Learning paradigm to train a generator G, a discriminator D and a surrogator F with only image-level supervision. During online inference, the generator G makes a one-stage detection without proposals object directly from images, which is very fast.

In this pipeline, object proposals in an image are treated as instances to form a bag, whose labels are assigned from the corresponding image-level annotations. The objective is then to estimate the probability of the instances containing a given object. The above pipeline has two key drawbacks: Firstly, instances are considered independent when obtaining foreground proposals. The correlation among instances is typically ignored and the optimization might converge to an undesirable local minimum [2]. Secondly, the existing methods follow a two/multi-stage process in detection, making the corresponding inference inefficient. Particularly, it requires to first extract object proposals [46, 55, 3], and thus cannot be applied to real-time applications. In addition, the image is typically augmented with five scales and horizontal flips [8, 28] in both training and testing, which further increases the online computation complexity significantly, sometimes to an order of magnitude.

In order to overcome the above limitations, in this paper, we train fast object detector for online WSD. Our goal is to achieve comparable speed to the state-of-the-art fast fully supervised detectors such as YOLO [35] or SSD [31], with comparable (or even better) detection accuracy. In particular, we utilize one-stage process in online detection, which discards the necessity of object proposals. We achieve this goal by proposing a novel Generative Adversarial Learning (GAL) paradigm, termed Generative Adversarial Learning Towards Fast Weakly Supervised Detection (GAL-fWSD), which regards the inference of WSD as a generative process, supervised by a discriminator. In particular, a discriminator D learns to distinguish the "real" distribution of bounding boxes from the "fake" distribution, while a generator G learns to fool D by detecting high-quality bounding boxes from images. Two specific designs are introduced:

First, since WSD does not have annotations of the ground-truth bounding boxes, we introduce a surrogator module F to mine promising bounding boxes from training data with only image-level annotations. It addresses the difficulty of modeling individual instances with only global labels. This pivotal module aggregates responses of individual object proposals to estimate a probabilistic distribution, which is then used to refine the proposals to train G.

Second, to address the drawback of low efficiency, the learned generator G is directly leveraged as a one-stage detector. To the best of our knowledge, this is the first attempt to learn genuine one-stage object detectors in a weakly supervised setting. As known in object detectors like SSD [31] and YOLO [35], one-stage detection is faster and simpler, but requires object-level annotations for training, which is traditionally infeasible in WSD.

The proposed GAL-fWSD framework is shown in Fig. 1. It works by iteratively learning a surrogator F to mine the object proposals of foreground categories, as well as learning a one-stage detector G to emulate F. The above procedure is supervised by a discriminator D. Besides the adversarial loss, we further introduce a structural similarity loss to encourage G to not only fool the discriminator D but also replicate the output of surrogator F, as inspired by GAN based image-to-image translation [24, 29, 53]. Meanwhile, the F module also augments its proposals with those generated by G to provide high-quality estimation of the distribution for ground-truth bounding boxes. At the test time, we only use the learned detector G for inference.

The contributions of this work are three folds:

- We propose a framework to adopt Generative Adversarial Learning to train fast detectors with only image-level annotation. To the best of our knowledge, this is the first attempt to formulate WSD using GAL instead of the traditional MIL paradigm.
- We propose to introduce a one-stage detector in WSD, instead of the traditional two/multi-stage detector. We achieve significant runtime speedup while maintaining comparable or better detection accuracy.
- To overcome the unavailability of ground-truth bounding boxes in training, we further propose to combine a structural similarity loss with an adversarial loss to train the generator network.

We present detailed evaluations on PASCAL VOC 2007, 2010 and 2012 [16], with comparison to several state-of-the-art methods [8, 28, 45, 30, 27, 15]. Experiment results demonstrate that our method processes an image of 300×300 size with astounding 118 frames per second (*i.e.*, up to $438\times$ faster than the state-of-the-art WSD in the literature), while still surpassing the detection accuracy of above methods. Using a larger 512×512 input, our method achieves even better accuracy, while can still perform real-time detection.

2. Related Work

Weakly Supervised Detection. Weakly supervised object detection has been widely studied in the past decade, which typically uses a two/multi-stage pipeline, *i.e.*, object proposal generation, feature extraction and proposal classification. Cinbis *et al.* [10] presented a multi-fold multiple instance learning approach, which avoids the performance degeneration in object localization. Wang *et al.* [47] utilized a probabilistic Latent Semantic Analysis (pLSA) to learn latent categories. The category containing target object class is selected by evaluating the discrimination score of each latent category. Bilen *et al.* [7] proposed to couple a smooth discriminative learning procedure with a convex clustering algorithm, which searches for a small set of exemplars to describe training data. Wang *et al.* [49] relaxed the highly combinatorial MIL optimization problem into a convex program and optimized it using stochastic gradient descent. Bazzani *et al.* [5] masked out regions of an image provided as input to a deep neural network, then embedded the drop of recognition score caused by masking out into an agglomerative clustering, which was used to merge regions for object localization.

In recent years, many WSD methods [8, 28, 45] have been proposed to learn end-to-end models with object proposals extracted from images. Bilen *et al.* [8] proposed a two-stream end-to-end CNN architecture. Kantorov *et al.* [28] further proposed to add a contrast-based contextual stream to form a three-stream CNN architecture. Tang *et al.* [45] integrated the basic multiple instance detection and multi-stage instance classification into a single network. Two-stage fully supervised detectors are also well investigated in weakly supervised setting [30, 27, 15]. Li *et al.* [30] presented a progressive domain adaptation approach

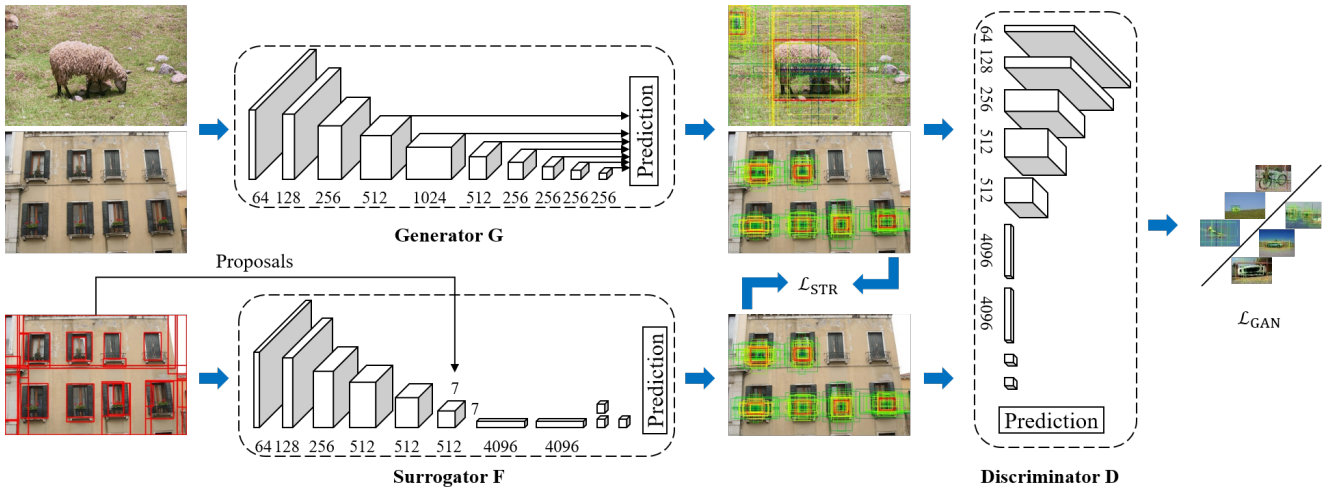

Figure 2: The proposed framework contains three modules during training. First, the generator G generates object bounding boxes in the given images. Second, the surrogator F estimates the distributions of bounding boxes from the given images and the corresponding image-level annotations. Third, the discriminator D distinguishes whether the images and distributions of bounding boxes are "real" (which is based on surrogator F) or "fake" (which comes from G). During inference, the generator G is used to detect objects one-stage from the input image.

with both classification adaptation and detection adaptation. Jie *et al.* [27] also learned a multi-label classification at the first step, followed by online supportive sample harvesting (augmented with a relative CNN score improvement metric) to detect object proposals to learn Fast RCNN [17]. Diba *et al.* [15] utilized a three-stage cascade model that incorporates localization, multiple instance learning and weakly supervised object segmentation to mine foreground proposals.

All above WSD methods require object proposals [46, 55, 3] to be generated in online inference. Despite such two-stage setting, some approaches [52, 54, 20, 50] proposed to localize objects without using explicit proposals. In those methods, response (or activation) map for each category was computed and then converted to a binary map, base on which bounding boxes are extracted. However, it is hard to estimate to what extend the objects are occluded mutually. Further more, those methods are still multi-stage detectors, as they require extra steps to extract bounding boxes. To our knowledge, no previous work has explored adapting one-stage detectors for WSD, since such one-stage detection like SSD [31] and YOLO [35] all require high-quality ground-truth object annotation for offline training.

Fast Object Detection. To speedup online inference, a natural choice is to decrease the proposal-wise subnetwork and increase the shared convolutional subnetwork. For example, R-CNN [18] had no shared convolutional layer, which made it very slow. In contrast, Faster R-CNN [36] shared most of the convolutional layer, which largely speedups from R-CNN. R-FCN [12] further removed all proposal-wise layers, which made it faster, but still required a time-consuming proposal feature pooling layer. Although SSD [31] had more than $8,000$ default boxes, it has neither proposal-wise layers nor proposal pooling layers, which makes it fastest among the above detectors. Institutionally, the network architecture also impacts the inference time. For example, Kantorov *et al.* [28] used Alex-Net [1] as backbone to reduce the network processing time. However, such a small network may hurt the accuracy.

Generative Adversarial Network. GAN was originally proposed by Goodfellow *et al.* [19] for synthesizing high-quality images. Recent efforts [9, 51, 33, 4, 21] are made to improve the stability of GAN in training. GAN has achieved impressive results in image generation [33], representation learning [37], style transfer [53] and image super-resolution [29]. However, there are very limited works that combine object detection and generative adversarial learning together. Wang *et al.* [48] utilized adversarial training to mine hard positive examples with different occlusions and deformations to train detectors. Contradictory from the above works, we adopt the generative adversarial learning to generate high-quality bounding boxes, which innovatively approximating the estimated distribution of bounding boxes for WSD.

3. The Proposed Method

3.1. Generative Adversarial Learning for Fast WSD

Typical fast one-stage object detectors are learned to fit the ground-truth bounding boxes of object instances. When turning to WSD, the annotation of object bounding boxes is not unavailable, and what we have is only the image-level annotation. For fast WSD, one usual solution is to train the object detector to fit the estimated bounding boxes to approximate the ground-truth ones. Nevertheless, conventional classification/regression losses [18, 17, 31, 34] are de-

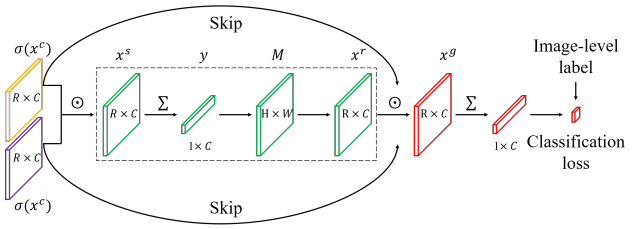

Figure 3: The network architecture of surrogator F.

fined on the ground-truth bounding boxes, and cannot capture the distribution p_b of the estimated bounding boxes.

Inspired by the recent work that used GAN to estimate generative models [19], we introduce an adversarial loss on the distribution of bounding boxes (described by location, size, and category), which enables the learning of detectors from only image-level annotations. As illustrate in Fig. 2, our key innovation lies in a novel Generative Adversarial Learning paradigm toward Fast Weakly Supervised Detection (GAL-fWSD). In the training stage, GAL-fWSD alternates between: (i) a discriminator D is introduced to distinguish the generated bounding boxes from the estimated ones, and (ii) the generator is updated to generate more high-quality bounding boxes to imitate the distribution p_b. Moreover, as explained in the following subsection, a structural similarity loss is further incorporated with adversarial loss to guide the learning of generator.

Another critical issue in GAL-fWSD is the estimation of bounding boxes based on image-level annotation. Here we present a surrogator F to estimate accurate proposals. In particular, the generator G is very fast without the need of object proposal, and it is our final detector to perform fast online detection, while the surrogator F is developed based on the slow but accurate proposals, both of which are mutually reinforced in the following Generative Adversarial Learning framework.

3.2. Model Architecture

Generator G. Give an image x, G outputs object bounding boxes $G^l(x)$ with the associated probability $G^c(x)$ and implicitly defines a probability distribution p_g. We utilize a one-stage proposal-free detector introduced in [31] as G. The reason of not using the more popular two-stage detectors is these detectors usually have an explicit object proposal stage, which either produces determined bounding boxes independent of object categories [46, 55, 3], or requires a complicated network structure [36]. The former approach restricts G from having sufficient diversity to model the distribution of bounding box p_b. And the latter approach closely couples proposal extractors with detectors, which makes the architecture complicated. On the contrary, one-stage detectors have the potential to be faster and simpler.

We briefly describe G below. In one-stage detectors, multiple feature maps at the top of network are used for prediction. Different feature map has different receptive fields, which allows to detect different scale of objects in the images. Each cell on feature map is associated to a set of default bounding boxes with different scales and aspect ratios implicitly. The detectors predict per-category scores of default bounding boxes, which indicate the presence of a category instance in each of those boxes. Then, regression is done to fit the bounding boxes tighter around the instance. Predictors in high-level feature maps are responsible for detecting large objects and vice versa, as illustrate in Fig. 2. More details can be found in [31].

Discriminator D. The discriminator D is designed to provide high-quality guidance to supervise G. Intuitively, it is a deep convolutional network whose inputs are an images x and bounding boxes $\{b_i\}$. In detail, VGG16 [41] is applied to obtain feature map $\phi(x)$ from its last convolution layer. Then the original fully-connected layers are adapted to take the feature map $\phi(x)$, normalized coordinates and probability of bounding boxes $\{b_i\}$ as inputs and output one entry, followed by a sigmoid layer to compute the probability that bounding boxes $\{b_i\}$ are high-quality for image x. This network is optimized to distinguish p_b and p_g using stochastic gradient descent with backpropagation.

Surrogator F. We use object-aware spatial information to refine the proposals, which is a variation of the method proposed in [8] by using object-aware spatial information to rectify proposals. As show in Fig. 3, F is a three-stream deep network. The proposal feature from the SPP layer [22] is forked into two streams, *i.e.*, classification stream and detection stream. Suppose we have C categories and R object proposals $\{p_1 \dots p_R\}$, the two streams produce two score matrices $x^c, x^d \in \mathbb{R}^{R\times C}$ by two fully-connected layers, respectively. Both score matrices are normalized by softmax functions over categories and proposals, respectively: $[\sigma(x^c)]_{ij} = \frac{e^{x^c_{ij}}}{\sum_{k=1}^{C} e^{x^c_{ik}}}$ and $[\sigma(x^d)]_{ij} = \frac{e^{x^d_{ij}}}{\sum_{r=1}^{R} e^{x^d_{rj}}}$. Then, the element-wise product of the output of the two streams is again a score matrix: $x^s = \sigma(x^c) \cdot \sigma(x^d)$. To acquire image-level classification scores, a sum pooling is further applied: $y_k = \sum_{r=1}^{R} x^s_{rk}$. Note x^s is calculated based on local information of each individual proposal. Therefore, the relationship between different proposals from the same image is abnegated and the learning process may converge to an undesirable local minimum. To handle this, a response map M^k for category k is obtained by back-propagating the image-level classification score y_k to pixels. As the image-level classification score is the sum of individual proposal scores, it is equivalent to back-propagating the score of each proposal to image pixels, which forms response maps that reveal the shape of objects.

To generate the output of the third stream, a rectifier is further defined based on the contrast between responses of a proposal and its contextual region: $x^r_{rk} = \rho_{rk} - \rho^c_{rk}$, where the first term ρ_{rk} estimates the response of each pro-

posal: $\rho_{rk} = \frac{1}{\sqrt{|p_r|}} \sum_{i,j \in p_r} \mathbb{1}[M_{ij}^k \geq \gamma \cdot \max M^k]$, and the second term ρ_{rk}^c estimates the contextual response of each proposal as ρ_{rk}.

The final score of each proposal is obtained by taking the element-wise product on the output of the three streams: $x^g = \sigma(x^c) \cdot \sigma(x^d) \cdot x^r$. Thus, the estimation x^s of the original proposal score is rectified by the output of the third stream x^r, leading to x^g.

3.3. The Model Objective

We combine the adversarial objective function L_{GAN} with a structural similarity function L_{STR}:

$$\mathcal{L} = \mathcal{L}_{\text{GAN}} + \lambda \mathcal{L}_{\text{STR}}, \tag{1}$$

where λ is the weight associated with second loss term.

Adversarial Loss. The objective function of generative adversarial learning is formulated as:

$$\min_G \max_D \mathcal{L}_{\text{GAN}} = \sum_{x,b \sim p_{\text{data}}(x,b)} \Big[\log \Big(D(x,b) \Big) \Big] + \sum_{x \sim p_{\text{data}}(x)} \Big[\log \Big(1 - D\big(x, G(x)\big) \Big) \Big], \tag{2}$$

where $p_{\text{data}}(x,b)$ denotes the bivariate distribution of images and the corresponding object bounding boxes. In particular, we consider a discriminator D (to distinguish which distribution the bounding boxes come from), and a generator G (to provide bounding boxes to fool the discriminator D). The distribution of object bounding boxes $p_{\text{data}}(x,b)$ is estimated by the surrogator F from the distribution of object proposals p. Such distribution can be obtained by running the existing algorithms such as [46, 55, 3]. We further have $b = F(x,p)$ in Eq. 2. It's worth noting that, although a bounding box b and an object proposal p may have the same coordinate, b is associated with a probabilistic distribution among multiple categories while p is only related with a category-less object proposal. To this end, we separate the objective function $\mathcal{L}_{\text{GAN}}$ to optimize the D and G, respectively:

$$D^* = \arg\max_D \sum_{x,p \sim p_{\text{data}}(x,p)} \Big[\log \Big(D\big(x, F(x,p)\big) \Big) \Big] + \sum_{x \sim p_{\text{data}}(x)} \Big[\log \Big(1 - D\big(x, G(x)\big) \Big) \Big], \tag{3}$$

$$G^* = \arg\min_G \sum_{x \sim p_{\text{data}}(x)} \Big[\log \Big(1 - D\big(x, G(x)\big) \Big) \Big]. \tag{4}$$

For each object category, the surrogator F aims to translate a distribution from p_p to p_b. If p_b approximates the ground-truth distribution well, the discriminator is able to provide high-quality supervision to improve G.

The above formulation encourages the detector, *i.e.*, generator G, to match the distribution of object bounding boxes. It differs from previous works that forced the detectors to fit just one discriminative bounding box per instance. In contrast, the generator G implicitly defines a probability distribution p_g. Then, G and D will reach a joint optimal [19] when the generative model perfectly replicates the bounding boxes distribution p_b, *i.e.*, $p_b = p_g$ [1] .

Structural Similarity Loss. We further introduce a structural similarity loss, which forces G to model not only p_b of the entire data, but also the bounding box patterns of individual images. We first match each bounding box generated by G to the bounding boxes from F with the best Jaccard overlap. For k-th category, $W_{ij}^k = \{1, 0\}$ is defined as the indicator for matching the i-th bounding boxes from G to the j-th from F. If the category of highest confidence score of bounding boxes from F is the background, the matched bounding boxes from G are treated as negative samples, and vice versa. The corresponding similarity function $\mathcal{L}_{\text{STR}}$ is defined as:

$$\sum_{x,p \sim p_{\text{data}}(x,p)} \Big[\mathcal{L}_{\text{conf}}\big(F^c(x,p), G^c(x)\big) + \alpha \mathcal{L}_{\text{loc}}\big(F^c(x,p), F^l(x,p), G^l(x)\big) \Big], \tag{5}$$

where $F^c(x,p)$ and $F^l(x,p)$ are the probability and coordinates of $F(x,p)$, respectively, which are similar for $G^c(x,z)$, $G^l(x)$ and $G(x)$. The first term of Eq. 5 measures the probability similarity of bounding boxes between $G(x)$ and $F(x,p)$, which is specified as:

$$\mathcal{L}_{\text{conf}}(T,S) = -\sum_{i=1} \sum_{j \in \text{Pos}} \sum_{k=1} T_{ik} W_{i,j}^k \log(S_{jk}) - \sum_{i=1} \sum_{j \in \text{Neg}} \sum_{k=1} (1 - T_{i0}) \log(S_{j0}), \tag{6}$$

where T_{ik} and T_{i0} denote the estimated probabilities of the i-th bounding box for category k and the background, respectively, which are similar for S_{jk} and S_{j0}.

The second term in Eq. 5 is localization similarity loss:

$$\mathcal{L}_{\text{loc}}(T,U,V) = \sum_{i=1} \sum_{j \in \text{Pos}} \sum_{k=1} T_{ik} W_{ij}^k \text{smooth}_{L1}(U_i - V_j), \tag{7}$$

where $U_i = \{U_i^x, U_i^y, U_i^w, U_i^h\}$ denotes the coordinates of the i-th bounding box, similar to V_j. The function $\text{smooth}_{L1}(Z)$ is similar to the regression loss as in [17]:

$$\text{smooth}_{L1}(Z) = \sum_{m \in \{x,y,w,h\}} \begin{cases} 0.5(Z^m)^2 & \text{if } |Z^m| < 1 \\ |Z^m| - 0.5 & \text{otherwise} \end{cases}. \tag{8}$$

[1] Note that a distortion operation is applied to images, which increases the diversity of bounding boxes from the generator G. As simulated to human annotators, we expect the generator to be able to produce notably different object bounding boxes given an image.

The weighting term α is set to 1 as in [31].

Then the full objective function can be reformulated as:

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{\text{GAN}} + \lambda\mathcal{L}_{\text{STR}} \\ &= \sum_{x,p\sim p_{\text{data}}(x,p)} \Big[\log\Big(D\big(x,F(x,p)\big)\Big) + \\ &\qquad \lambda\Big\{\mathcal{L}_{\text{conf}}\big(F^c(x,p),G^c(x)\big) + \\ &\qquad \alpha\mathcal{L}_{\text{loc}}\big(F^c(x,p),F^l(x,p),G^l(x)\big)\Big\}\Big] + \\ &\qquad \sum_{x\sim p_{\text{data}}(x)} \Big[\log\Big(1-D\big(x,G(x)\big)\Big)\Big].\end{aligned} \tag{9}$$

Feedback Mechanism. The surrogator F is not fixed during training, which is instead updated iteratively with D and G. At each step, we first update D, then G, similar to the vanilla GAN [19]. To update F, we first merge the original objectness proposals with the bounding boxes generated by G, then F is updated with the augmented proposals. Therefore, all function $F(x,p)$ in Eq. 9 are reformulated as $F(x,\{p,G^l(x)\})$. Although the bounding boxes generated by G are category-specific, we treat them as object proposal for simplicity.

During training, G feedbacks high-quality bounding boxes to improve the performance of F, which in turn provides high-quality estimation of distribution p_b to improve G through $\mathcal{L}_{\text{GAN}}$ and $\mathcal{L}_{\text{STR}}$ (implicitly or explicitly). With such feedback, G is capable to learn unknown distribution of bounding boxes, which is estimated by F from an alterable set of object proposals. If G generates high-quality bounding boxes, $G^l(x)$ may dominate the distribution of $F(x,\{p,G(x)\})$, *i.e.*, average of $F^c(x,G(x))$ is higher than that of $F^c(x,p)$. From this perspective, the G also fools F, as G provides higher-quality bounding boxes than the original proposals.

4. Experiment

4.1. Experimental Setup

Datasets. We conduct experiments on PASCAL VOC 2007, 2010 and 2012 [16], which are commonly used in object detection. PASCAL VOC 2007 consists of $2,501$ training images, $2,510$ validation images, and $4,092$ test images over 20 categories. PASCAL VOC 2010 consists of $4,998$ training images, $5,105$ validation images, and $9,637$ test images over 20 categories. PASCAL VOC 2012 consists of $5,717$ training images, $5,823$ validation images, and $10,991$ test images over 20 categories. We use both train and validation splits as our training sets, and the test split as our test set. As we focus on weakly supervised detection, only image-level labels are used in training.

Evaluation Protocols. First, we evaluate the mean average precision (mAP) on the test set following the standard PASCAL VOC protocol [16]. Second, because PASCAL VOC actually has object-level annotation, we compute CorLoc [14] on the training set. CorLoc is a metric to evaluate object localization. It computes the percentage of images in which a method correctly localizes an object of the target category. Following [16], a bounding box is considered correct if it has at least 0.5 intersection-over-union with any ground truth bounding boxes of the target category.

Implementation Details. We use VGG16 [41] pre-trained on ImageNet [13] as the backbone for all modules, *i.e.*, D, G and F. Pre-training on ImageNet classification data [13] does not require bounding box annotations. We set the parameter $\lambda = 10^{-3}$. To obtain a good initialization for G, we first train F with the original objectness proposals from [3] with a learning rate of 10^{-3}. Then we iteratively update the above models. At each step, we train D for $1,000$ iterations using stochastic gradient descent with a learning rate of 10^{-3}, momentum of 0.9, weight decay of 0.005, minibatch size of 128, while a minibatch size of 32 is used to train G. To train F, we keep the same setting as D for 40 iterations, which is about a epoch of VOC 2007 training set. We set the rest hyper-parameters of G and F following respective papers [31, 8]. To trade-off speed and accuracy, we use two difference input sizes: 300×300 and 512×512 for G. Our implementation is done using Caffe [25]

4.2. Comparison to State-of-the-Arts

Tab. 1 shows our results on PASCAL VOC 2007 test split in termed of mAP. The low-resolution version, *i.e.*, GAL-fWSD300, is already more accurate than the state-of-the-art methods. Note that 300×300 is the lowest resolution of input in Tab. 1. When we train GAL-fWSD on a larger 512×512 input size, it further surpasses the state-of-the-arts [8] and [15] by 12.7% and 4.7% mAP, respectively. We want to point out that most methods in Tab. 1 used a multi-scale setting during training and testing, and their input resolution of the max scale is far larger than ours. Our performance can be further improved by using the context information [28], size estimation [39] and the ensemble of multiple networks [44]. GAL-fWSD shows large improvement for categories with deformable parts, for example, *bird*, *dog*, *cat*, and *person*. All alternative methods did not perform well in the *person* category, whose mAPs are all $\leq 20.3\%$. Our GAL-fWSD300 achieves 81.3% improvement compared to [11] for *person* category. However, our modes are not good at detecting *horse*. By using the detection analysis tool [23], we find that most false positives are due to poor localization.

Tab. 2 shows our results on the PASCAL VOC 2007 training set in termed of CorLoc. Our two models consistently boost the performance by 11.4% and 9.4%, respectively, compared to [15]. It indicates that our model

Table 1: Detection comparisons to the state-of-the-art methods on PASCAL VOC 2007 test set in terms of AP(%).

Method		aero	bike	bird	boat	btl	bus	car	cat	chair	cow	table	dog	horse	mbike	pers	plant	sheep	sofa	train	tv	**Av.**
Song *et al.*'14	[43]	27.6	41.9	19.7	9.1	10.4	35.8	39.1	33.6	0.6	20.9	10.0	27.7	29.4	39.2	9.1	19.3	20.5	17.1	35.6	7.1	22.7
Wang *et al.*'14	[47]	48.9	42.3	26.1	11.3	11.9	41.3	40.9	34.7	10.8	34.7	18.8	34.4	35.4	52.7	19.1	17.4	35.9	33.3	34.8	46.5	31.6
Cinbis *et al.*'15	[11]	39.3	43.0	28.8	20.4	8.0	45.5	47.9	22.1	8.4	33.5	23.6	29.2	38.5	47.9	20.3	20.0	35.8	30.8	41.0	20.1	30.2
Bilen *et al.*'16	[8]	39.4	50.1	31.5	16.3	12.6	64.5	42.8	42.6	10.1	35.7	24.9	38.2	34.4	55.6	9.4	14.7	30.2	40.7	54.7	46.9	34.8
Bency *et al.*'16	[6]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	25.7
Li *et al.*'16	[30]	54.5	47.4	41.3	20.8	17.7	51.9	63.5	46.1	21.8	57.1	22.1	34.4	**50.5**	61.8	16.2	**29.9**	40.7	15.9	55.3	40.2	39.5
Kantorov *et al.*'16	[28]	57.1	52.0	31.5	7.6	11.5	55.0	53.1	34.1	1.7	33.1	49.2	42.0	47.3	56.6	15.3	12.8	24.8	48.9	44.4	47.8	36.3
Jie *et al.*'17	[27]	52.2	47.1	35.0	26.7	15.4	61.3	66.0	54.3	3.0	53.6	24.7	43.6	48.4	**65.8**	6.6	18.8	**51.9**	43.6	53.6	62.4	41.7
Tang *et al.*'17	[45]	58.0	62.4	31.1	19.4	13.0	65.1	62.2	28.4	**24.8**	44.7	30.6	25.3	37.8	65.5	15.7	24.1	41.7	46.9	64.3	**62.6**	41.2
Shi *et al.*'17	[38]	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	33.8
Diba *et al.*'17	[15]	49.5	60.6	38.6	**29.2**	16.2	**70.8**	56.9	42.5	10.9	44.1	29.9	42.2	47.9	64.1	13.8	23.5	45.9	**54.1**	60.8	54.5	42.8
GAL-fWSD300		52.0	60.5	44.6	26.1	20.6	63.1	**66.2**	**65.3**	15.0	50.1	**52.8**	**56.7**	21.3	63.4	**36.8**	22.7	47.9	51.7	68.9	54.1	**47.0**
GAL-fWSD512		**58.4**	**63.8**	**45.8**	24.0	**22.7**	67.7	65.7	58.9	15.0	**58.1**	47.0	53.7	23.8	64.3	36.2	22.3	46.7	50.3	**70.8**	55.1	**47.5**

Table 2: Localization comparisons to the state-of-the-art methods on PASCAL VOC 2007 training set in terms of CorLoc(%).

Method		aero	bike	bird	boat	btl	bus	car	cat	chair	cow	table	dog	horse	mbike	pers	plant	sheep	sofa	train	tv	**Av.**
Siva *et al.*'12	[42]	45.8	21.8	30.9	20.4	5.3	37.6	40.8	51.6	7.0	29.8	27.5	41.3	41.8	47.3	24.1	12.2	28.1	32.8	48.7	9.4	30.2
Shi *et al.*'13	[40]	67.3	54.4	34.3	17.8	1.3	46.6	60.7	68.9	2.5	32.4	16.2	58.9	51.5	64.6	18.2	3.1	20.9	34.7	63.4	5.9	36.2
Wang *et al.*'14	[47]	80.1	63.9	51.5	14.9	21.0	55.7	74.2	43.5	26.2	53.4	16.3	56.7	58.3	69.5	14.1	38.3	58.8	47.2	49.1	60.9	48.5
Cinbis *et al.*'15	[11]	65.3	55.0	52.4	48.3	18.2	66.4	77.8	35.6	26.5	67.0	46.9	48.4	70.5	69.1	35.2	35.2	69.6	43.4	64.6	43.7	52.0
Wang *et al.*'15	[49]	37.7	58.8	39.0	4.7	4.0	48.4	70.0	63.7	9.0	54.2	33.3	37.4	61.6	57.6	30.1	31.7	32.4	52.8	49.0	27.8	40.2
Bilen *et al.*'16	[8]	65.1	58.8	58.5	33.1	39.8	68.3	60.2	59.6	34.8	64.5	30.5	43.0	56.8	82.4	25.5	41.6	61.5	55.9	65.9	63.7	53.5
Li *et al.*'16	[30]	78.2	67.1	61.8	38.1	36.1	61.8	78.8	55.2	28.5	68.8	18.5	49.2	64.1	73.5	21.4	47.4	64.6	22.3	60.9	52.3	52.4
Kantorov *et al.*'16	[28]	83.3	68.6	54.7	23.4	18.3	73.6	74.1	54.1	8.6	65.1	47.1	59.5	67.0	83.5	35.3	39.9	67.0	49.7	63.5	65.2	55.1
Jie *et al.*'17	[27]	72.7	55.3	53.0	27.8	35.2	68.6	81.9	60.7	11.6	71.6	29.7	54.3	64.3	88.2	22.2	53.7	72.2	52.6	68.9	75.5	56.1
Zhu *et al.*'17	[54]	**85.3**	64.2	**67.0**	42.0	16.4	71.0	64.7	**88.7**	20.7	63.8	58.0	**84.1**	**84.7**	80.0	60.0	29.4	56.3	68.1	77.4	30.5	60.6
Tang *et al.*'17	[45]	81.7	80.4	48.7	**49.5**	32.8	**81.7**	85.4	40.1	**40.6**	**79.5**	35.7	33.7	60.5	**88.8**	21.8	**57.9**	76.3	59.9	75.3	**81.4**	60.6
Shi *et al.*'17	[38]	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	59.5
Diba *et al.*'17	[15]	83.9	72.8	64.5	44.1	40.1	65.7	82.5	58.9	33.7	72.5	25.6	53.7	67.4	77.4	26.8	49.1	68.1	27.9	64.5	55.7	56.7
GAL-fWSD300		76.5	76.1	64.2	48.1	**52.5**	80.7	**86.1**	73.9	30.8	78.7	**62.0**	71.5	46.7	86.1	**60.7**	47.8	**82.3**	**74.7**	83.1	79.3	**68.1**
GAL-fWSD512		78.6	**81.9**	63.6	40.3	48.8	80.7	85.3	76.3	30.3	78.0	54.5	65.3	48.4	86.5	56.3	46.9	76.0	68.1	**83.9**	73.1	**66.1**

Table 3: Ablation study of GAL-fWSD300 on Pascal VOC.

(a) Effects of various components.

$\mathcal{L}_{\text{GAN}}$	$\mathcal{L}_{\text{STR}}$	#Feedback	mAP
✓			39.2
✓	✓		44.3
✓	✓	128	45.8
✓	✓	256	46.5
✓	✓	512	47.0
✓	✓	1024	47.1

(b) Feedback proposals *vs.* original proposals.

#Proposals	512		1536	
+Feedback		+512		+512
+Original	+512		+512	
Total	1024		2048	
mAP	43.8	46.5	44.3	47.0

Table 4: Detection and localization comparisons to the state-of-the-art methods on PASCAL VOC 2010 and 2012 in terms of mAP (%) and CorLoc (%). †http://host.robots.ox.ac.uk:8080/anonymous/2TGWVW.html §http://host.robots.ox.ac.uk:8080/anonymous/5XCKR0.html

Method		2010		2012	
		mAP (%)	CorLoc (%)	mAP (%)	CorLoc (%)
Oquab *et al.*'15	[32]	–	–	11.7	–
Cinbis *et al.*'15	[11]	27.4	55.2	–	–
Li *et al.*'16	[30]	30.7	–	29.1	–
Kantorov *et al.*'16	[28]	–	–	35.3	54.8
Bency *et al.*'16	[6]	–	–	26.5	–
Tang *et al.*'17	[45]	–	–	37.9	62.1
Jiang *et al.*'17	[26]	36.0	–	33.6	–
Diba *et al.*'17	[15]	39.5	–	37.9	–
GAL-fWSD300		45.1†	68.3	43.1§	67.2

achieves the best localizing performance (6 out of 20 categories in GAL-fWSD300), which verifies the effectiveness of our scheme from another perspective.

We further conduct experiments on PASCAL VOC 2010 and 2012. Tab. 4 shows our result on the PASCAL VOC 2010 and 2012 on both metrics. Compared to other methods, our method consistently achieves the state-of-the-art performance using a single model. We further highlight that, even using low-resolution input size (300×300), our method still significantly outperforms all methods that use multi-scale input for both training and testing.

Table 5: Efficiency comparisons to the state-of-the-art methods on PASCAL VOC 2007 test set in terms of speed(FPS).

Method		Backbone	# Scales	Flips?	Time(ms)				FPS
					Proposal	Network	NMS	Total	
Bilen *et al.*'16	[8]	VGG16	5	✓	250	3451	19	3720	0.27
Li *et al.*'16	[30]	VGG16	1	✗	250	217	6	473	2.11
Kantorov *et al.*'16	[28]	VGG-F	5	✓	1510	1140	14	2664	0.38
Tang *et al.*'17	[45]	VGG16	5	✓	1510	2052	4	3566	0.28
Jie *et al.*'17	[27]	VGG16	1	✗	250	321	6	577	1.73
GAL-fWSD300		VGG16	1	✗	0	8.28	0.2	8.48	118
GAL-fWSD512		VGG16	1	✗	0	19.73	0.2	19.93	50

4.3. Online Inference Efficiency

We compare our method to several state-of-the-art methods [8, 30, 28, 45, 27] in term of inference time. For fair comparison, we re-implement the methods in [8, 28, 27] with Caffe [25] framework. Thus, the time costs of all methods in Tab. 5 are comparable. We reproduce all methods in Tab. 5 on our server and keep all the other settings exactly the same. We measure the speed using a GTX 1080Ti GPU and cuDNN v6 with Intel i7-6900K@3.20GHz.

Tab. 5 shows the comparison between GAL-fWSD and other the state-of-the-art methods. Our fastest model (GAL-fWSD300) quantitatively performs at 118 frame per second and is $55\times$ and $438\times$ speedup comparing to [30] and [8], respectively. According to the profiling, there are three main reasons that make detectors inefficient: First, the multi-

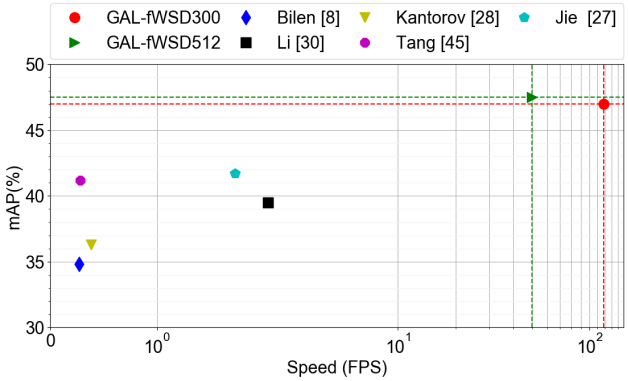

Figure 4: Speed and accuracy comparison between different WSD methods on PASCAL VOC 2007 detection task. The mAPs are from the respective paper, although our reproduction results may have slight difference.

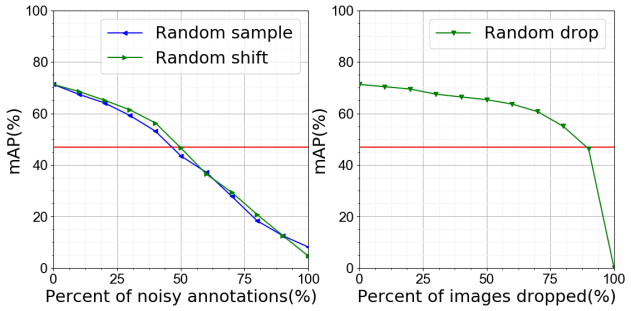

Figure 5: The performance of SSD300 with various annotations. The red line shows the performance of GAL-fWSD300.

scale setting in inference significantly pulls down the speed [8, 28, 45]. Second, the proposal extraction used at least 250 milliseconds per image, as illustrated in the fifth column in Tab. 5. Third, the proposal-wise operation in network, *e.g.* SPP layer, also consumes a lot of time. Fig. 4 further demonstrates that our both methods outperform all other in terms of speed and accuracy by a large gap.

4.4. Ablation Study

We use GAL-fWSD300 on Pascal VOC 2007 test set for ablation study. As shown in Tab. 3a, optimizing $\mathcal{L}_{\text{GAN}}$ and $\mathcal{L}_{\text{STR}}$ jointly improves the performance by 5.1%, which shows the effectiveness of the structural similarity loss. With the feedback mechanism, the performance is further improved by a large margin, which confirms our intuition that feedback proposals can be used to help surrogator F, and in turn boost the performance of generator G. Tab. 3b also shows that feedback mechanism is more effective than simply increasing the number of region proposals on improving the performance.

4.5. Robustness to Noisy Annotation

We study the robustness of G by analyzing the performance of SSD300 [31] when applying synthetic noise to the ground-truth annotations. The models are trained on PASCAL VOC 2007 training and validation split, and tested on the test split. We introduce two mechanisms to emulate noise. The first replaces the ground-truth bounding boxes by randomly sampling a region from the images. The second randomly shifts the ground-truth bounding boxes, but keeps the centers of bounding boxes inside the image boundary. As illustrated in the left of Fig. 5, with the increasing proportion of noisy annotations, the performance of G drops dramatically. It reveals that we need a precise estimation of the bounding boxes distribution to supervise the training of G.

We also randomly drop a proportion of training image to study the robustness of G as illustrated in the right of Fig. 5, which indicates the importance of annotations amount, *i.e.*, ground-truth bounding boxes. The red line in Fig. 5 indicates the performance of GAL-fWSD300. Our detection accuracy approximates the fully supervised methods, whose training data contains about $46\% \sim 50\%$ noisy annotations. The performance of SSD300 drops to GAL-fWSD300 when it has only 10% training images. In other words, an image with object-level annotations (which are required by fully supervised detectors) is probably worth 10 images with only image-level annotations in our method.

5. Conclusion

In this paper, we propose an effective framework **GAL-fWSD** towards real-time weakly supervised object detection, which tackles the need of bounding-box-level supervision from a novel perspective of Generative Adversarial Learning. In particular, we design the generator to be extremely fast by using state-of-the-art one-stage object detector. In addition, a surrogator module and feedback mechanism are introduced to estimate the distribution of object bounding boxes. Finally, a novel structural similarity loss together with adversarial loss is further proposed to optimize the model. Extensive experiments show that GAL-fWSD significantly speedups the state-of-the-art weakly supervised detectors while achieving the state-of-the-art detection accuracy.

6. Acknowledgment

This work is supported by the National Key R&D Program (No.2017YFC0113000, and No.2016YFB1001503), Nature Science Foundation of China (No.U1705262, No.61772443, and No.61572410), Post Doctoral Innovative Talent Support Program under Grant BX201600094, China Post-Doctoral Science Foundation under Grant 2017M612134, Scientific Research Project of National Language Committee of China (Grant No. YB135-49), and Nature Science Foundation of Fujian Province, China (No. 2017J01125).

References

[1] K. Alex, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[2] J. Amores. Multiple instance classification: Review, taxonomy and comparative study. *Artificial Intelligence*, 2013.

[3] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale Combinatorial Grouping. In *CVPR*, 2014.

[4] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. In *ArXiv*, 2017.

[5] L. Bazzani, A. Bergamo, D. Anguelov, and L. Torresani. Self-Taught Object Localization with Deep Networks. In *WACV*, 2016.

[6] A. J. Bency, H. Kwon, H. Lee, S. Karthikeyan, and B. S. Manjunath. Weakly Supervised Localization using Deep Feature Maps. In *ECCV*, 2016.

[7] H. Bilen, M. Pedersoli, and T. Tuytelaars. Weakly supervised object detection with convex clustering. In *CVPR*, 2015.

[8] H. Bilen and A. Vedaldi. Weakly Supervised Deep Detection Networks. In *CVPR*, 2016.

[9] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In *NIPS*, 2016.

[10] R. G. Cinbis, J. Verbeek, and C. Schmid. Multi-fold MIL Training for Weakly Supervised Object Localization. In *CVPR*, 2014.

[11] R. G. Cinbis, J. Verbeek, and C. Schmid. Weakly Supervised Object Localization with Multi-fold Multiple Instance Learning. *TPAMI*, 2015.

[12] J. Dai, Y. Li, K. He, and J. Sun. R-FCN: Object Detection via Region-based Fully Convolutional Networks. In *NIPS*, 2016.

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

[14] T. Deselaers, B. Alexe, and V. Ferrari. Weakly supervised localization and learning with generic knowledge. *IJCV*, 2012.

[15] A. Diba, V. Sharma, A. Pazandeh, H. Pirsiavash, and L. Van Gool. Weakly Supervised Cascaded Convolutional Networks. In *CVPR*, 2017.

[16] M. Everingham, L. Van˜Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results.

[17] R. Girshick. Fast R-CNN. In *ICCV*, 2016.

[18] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[19] I. J. Goodfellow, J. Pouget-abadie, M. Mirza, B. Xu, and D. Warde-farley. Generative Adversarial Nets. In *NIPS*, 2014.

[20] A. Gudi, N. van Rosmalen, M. Loog, and J. van Gemert. Object-Extent Pooling for Weakly Supervised Single-Shot Localization. In *BMVC*, 2017.

[21] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved Training of Wasserstein GANs. In *ArXiv*, 2017.

[22] K. He, X. Zhang, S. Ren, and J. Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In *ECCV*, 2014.

[23] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing Error in Object Detectors. In *ECCV*, 2012.

[24] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. In *CVPR*, 2017.

[25] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, and U. C. B. Eecs. Caffe : Convolutional Architecture for Fast Feature Embedding. In *ACMMM*, 2014.

[26] W. Jiang, T. Ngo, B. S. Manjunath, Z. Zhao, and F. Su. Optimizing Region Selection for Weakly Supervised Object Detection. *ArXiv*, 2017.

[27] Z. Jie, Y. Wei, X. Jin, J. Feng, and W. Liu. Deep Self-Taught Learning for Weakly Supervised Object Localization. In *CVPR*, 2017.

[28] V. Kantorov, M. Oquab, M. Cho, and I. Laptev. ContextLocNet: Context-Aware Deep Network Models for Weakly Supervised Localization. In *ECCV*, 2016.

[29] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In *CVPR*, 2017.

[30] D. Li, J.-B. Huang, Y. Li, S. Wang, and M.-H. Yang. Weakly Supervised Object Localization with Progressive Domain Adaptation. In *CVPR*, 2016.

[31] 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.

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

[33] A. Radford, L. Metz, and S. Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *ICLR*, 2016.

[34] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *CVPR*, 2016.

[35] J. Redmon and A. Farhadi. YOLO9000: Better, Faster, Stronger. In *CVPR*, 2017.

[36] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NIPS*, 2015.

[37] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved Techniques for Training GANs. In *NIPS*, 2016.

[38] M. Shi, H. Caesar, and V. Ferrari. Weakly Supervised Object Localization Using Things and Stuff Transfer. In *ICCV*, 2017.

[39] M. Shi and V. Ferrari. Weakly Supervised Object Localization Using Size Estimates. In *ECCV*, 2016.

[40] Z. Shi, T. M. Hospedales, and T. Xiang. Bayesian Joint Topic Modelling for Weakly Supervised Object Localisation. In *ICCV*, 2013.

[41] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015.

[42] P. Siva, C. Russell, and T. Xiang. In defence of negative mining for annotating weakly labelled data. In *ECCV*, 2012.

[43] H. O. Song, R. Girshick, S. Jegelka, J. Mairal, Z. Harchaoui, and T. Darrell. On learning to localize objects with minimal supervision. In *ICML*, 2014.

[44] W. R. Tan, C. S. Chan, H. Aguirre, and K. Tanaka. ArtGAN: Artwork Synthesis with Conditional Categorial GANs. *Arxiv*, 2017.

[45] P. Tang, X. Wang, X. Bai, and W. Liu. Multiple Instance Detection Network with Online Instance Classifier Refinement. In *CVPR*, 2017.

[46] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective Search for Object Recognition. *IJCV*, 2013.

[47] C. Wang, W. Ren, K. Huang, and T. Tan. Weakly Supervised Object Localization with Latent Category Learning. In *ECCV*, 2014.

[48] X. Wang, A. Shrivastava, and A. Gupta. A-Fast-RCNN: Hard Positive Generation via Adversary for Object Detection. In *CVPR*, 2017.

[49] X. Wang, Z. Zhu, C. Yao, and X. Bai. Relaxed Multiple-Instance SVM with Application to Object Discovery. In *ICCV*, 2015.

[50] S. Yunhang, J. Rongrong, W. Changhu, L. Xi, and L. Xuelong. Weakly Supervised Object Detection via Object-Specific Pixel Gradient. *TNNLS*, 2018.

[51] J. Zhao, M. Mathieu, and Y. LeCun. Energy-based Generative Adversarial Network. In *ICLR*, 2016.

[52] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. In *CVPR*, 2016.

[53] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *ICCV*, 2017.

[54] Y. Zhu, Y. Zhou, Q. Ye, Q. Qiu, and J. Jiao. Soft Proposal Networks for Weakly Supervised Object Localization. In *ICCV*, 2017.

[55] L. Zitnick and P. Dolla. Edge Boxes:Locating Object Proposals from Edges. In *ECCV*, 2014.