NOTE-RCNN: NOise Tolerant Ensemble RCNN for Semi-Supervised Object Detection

Jiyang Gao[1] Jiang Wang[2] Shengyang Dai[2] Li-Jia Li[3] Ram Nevatia[1]
[1]University of Southern California [2]Google Could [3]Stanford University
{jiyangga, nevatia}@usc.edu, {wangjiang, sydai}@google.com, lijiali@cs.stanford.edu

Abstract

*The labeling cost of large number of bounding boxes is one of the main challenges for training modern object detectors. To reduce the dependence on expensive bounding box annotations, we propose a new semi-supervised object detection formulation, in which a few seed box level annotations and a large scale of image level annotations are used to train the detector. We adopt a training-mining framework, which is widely used in weakly supervised object detection tasks. However, the mining process inherently introduces various kinds of labelling noises: false negatives, false positives and inaccurate boundaries, which can be harmful for training the standard object detectors (*e.g. *Faster RCNN). We propose a novel NOise Tolerant Ensemble RCNN (NOTE-RCNN) object detector to handle such noisy labels. Comparing to standard Faster RCNN, it contains three highlights: an ensemble of two classification heads and a distillation head to avoid overfitting on noisy labels and improve the mining precision, masking the negative sample loss in box predictor to avoid the harm of false negative labels, and training box regression head only on seed annotations to eliminate the harm from inaccurate boundaries of mined bounding boxes. We evaluate the methods on ILSVRC 2013 and MSCOCO 2017 dataset; we observe that the detection accuracy consistently improves as we iterate between mining and training steps, and state-of-the-art performance is achieved.*

1. Introduction

With the recent advances in deep learning, modern object detectors, such as Faster RCNN [21], YOLO [20], SSD [19] and RetinaNet [17], are reliable in predicting both object classes and their bounding boxes. However, the application of deep learning-based detectors is still limited by the efforts of collecting bounding box training data. These detectors are trained with huge amount of manually labelled bounding boxes. In real world, each application may require us to detect a unique set of the categories. It's expensive and time-consuming to label tens of thousands of object bounding boxes for each application.

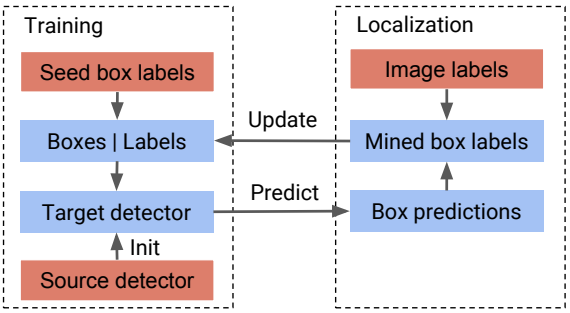

Figure 1. Iterative training-mining pipeline.

To reduce the effort of labelling bounding boxes, researchers worked on learning object detectors with only image-level labels, which are substantially cheaper to annotate, or even free with image search engines; this task is called weakly supervised object detection [4, 25, 29]. Multiple Instance Learning (MIL) [6] based training-mining pipeline [4, 25, 27] is widely used for this task; however, the resulting detectors perform considerably worse than the fully supervised counterparts. We believe the reasons are two-fold: First, a detector learned with only image-level labels often performs poorly in localization, it may focus on the object part, but not the whole object (*e.g.*, in Figure 2, "cat" detector detects cat head); second, without an accurate detector, object instances cannot be mined correctly, especially when the scene is complicated.

To address the aforementioned problems in weakly supervised object detection, we propose a semi-supervised object detection setting: learning an object detector with a limited amount of labelled bounding boxes (e.g. 10 to 20 images with fully labeled bounding boxes) as well as a large amount of image-level labels. Specifically, we want to train an object detector for a set of *target* categories. For target categories, a small amount of *seed* bounding box annotations and a large amount of image-level annotations are

available for training. We also assume that a pre-trained object detector for *source* categories is available. The source and target categories do not overlap with each other. Given the wide availability of large scale object detection datasets, such as MSCOCO [18] and ILSVRC [22], this assumption is not hard to satisfy in practice. This assumption is not essential for the formulation either. Note that our formulation is different from previous semi-supervised object detection [11, 26], in which the seed bounding box annotations are not considered.

Training-mining framework is widely utilized in weakly supervised object detection [25, 4, 27, 29]. The standard training-mining pipeline [4, 25] in weakly supervised object detection iterates between the following steps: 1. Train object detector with the mined bounding boxes (the initial detector is trained with the whole images and the labels); 2. Mine the bounding boxes with the current object detector. A straight-forward way to incorporate the seed bounding boxes is that we use them to train the initial object detector, mine bounding boxes with the initial object detector, train a new detector with both seed and mined bounding boxes, and iterate between mining and training steps.

The mining process inherently introduces various types of noise. First, mining process inevitably misses some objects, which are treated as negative (*i.e*. background) samples in training phase; such false negatives are harmful for training the classification head of object detector. Second, the boundaries of the mined bounding boxes are not precise, which is harmful for learning the box regression head of the detector. Third, the class labels of the mined boxes cannot be 100% accurate, leading to some false positives. Some visualization examples for the mined labels from the baseline method are shown in Figure 2. Because of these issues, we observe that the detection accuracy usually decreases as we iterate between training and mining steps if standard object detector architecture (*e.g*. Faster RCNN) is employed.

We propose a novel NOise Tolerant Ensemble RCNN (NOTE-RCNN) architecture. The NOTE-RCNN incorporates an ensemble of classification heads for both box predictor (*i.e*. second stage) and region proposal predictor (*i.e*. first stage) to increase the precision of the mined bounding boxes, *i.e*., reduce false positives. Specifically, one classification head is only trained with seed bounding box annotations; the other head is trained with both seed and mined box annotations. The consensus of the both heads is employed to determine the confidence of the classification. This is similar to recent work that uses ensemble for robust estimation of prediction confidence [3, 2]. We also utilize the knowledge of the pre-trained detector on source categories as *weak teachers*. Specifically, another classification head is added to distill knowledge [10] from a weak teacher; the distillation process acts as a regularizer to prevent the network from overfitting on the noisy annotations.

Figure 2. Top: examples of weakly supervised object detection failure cases: poor localization; objects can't be discovered in complicated scenes. Bottom: examples of the mined box noises using a standard faster RCNN: 1) false negatives, 2) false positives, 3) inaccurate box boundaries; groundtruth boxes are in black, mined boxes are in other colors.

The NOTE-RCNN architecture is also designed to be robust to false negative labels. For the classification head in the box predictor that uses mined bounding boxes for training, we remove the loss of predicting negatives (*i.e*. background) from its training loss, thus the training is not affected by the false negatives. Finally, the regression head is only trained with the seed bounding boxes, which avoids it being affected by the inaccurate boundaries of the mined bounding boxes.

We evaluated the proposed architecture on MSCOCO [18] and ILSVRC [22] datasets. The experimental results show that the proposed framework increases the precision of mined box annotations and can bring up to 40% improvement on detection performance by iterative training. Compared with weakly supervised detection, training with seed annotations using NOTE-RCNN improves the state-of-the-art performance from 36.9% to 43.7%, while using standard Faster RCNN only achieves 38.7%.

In summary, our contributions are three-fold: first, we propose a practical semi-supervised object detection problem, with a limited amount of labelled bounding boxes as well as a large amount of image-level labels; second, we identified three detrimental types of noise that inherently exists in training-mining framework ; third, we propose a novel NOTE-RCNN architecture that is robust to such noise, and achieves state-of-the-art performance on benchmark datasets.

2. Related Work

Weakly supervised object detection. The majority of recent work treats weakly supervised object detection as a Multiple Instance Learning (MIL) [6] problem. An image is decomposed into object proposals using proposal generators, such as EdgeBox [30] or SelectiveSearch [28]. The basic pipeline is to iteratively mine (*i.e*. localize) objects as

training samples using the detectors and then train detectors with the updated training samples. The detector can be a proposal level SVM classifier [23, 4, 27] or modern CNN-based detector [25, 14, 29], such as RCNN [9] or Fast RCNN [8]. Deselaers *et al*. [5] first argued to use objectness score as a generic object appearance prior to the particular target categories. Cinbis *et al*. [4] proposed a multi-fold multiple instance learning procedure, which prevents training from prematurely locking onto erroneous object locations. Uijlings *et al*. [27] argued to use pre-trained detectors as the proposal generator and show its effectiveness in knowledge transfer from source to target categories.

Recently, there are also work that designs end-to-end deep networks combining with multiple instance learning. Bilen *et al*. [1] designed a two-stream network, one for classification and the other for localization, it outputs final scores for the proposals by the element-wise multiplication on the scores from the two streams. Kantorov et al. *et al*. [15] proposed a context-aware CNN model based on contrast and additive contextual guidance, which improved the object localization accuracy.

Semi-supervised object detection. Note that in previous work [11], the definition of semi-supervised object detection is slightly different from ours, in which only the image-level labels and pre-trained source detectors are considered, but seed bounding box annotations are not used. Beginning from LSDA [11], Hoffman *et al*. proposed to learn parameter transferring functions between the classification network and the detection network, so that a classification model trained by image level labels can be transferred to a detection model. Tang *et al*. [26] explored the usage of visual and semantic similarities among the source categories and the target categories in the parameter transferring function. Hu *et al*. [12] further extended this method to semi-supervised instance segmentation, which transfers models for object detection to instance segmentation. Uijlings *et al*. [27] adopted the MIL framework from weakly supervised object detection, and replaced the unsupervised proposal generator [30] by the pre-trained source detectors to use the shared knowledge. Li *et al*. [16] proposed to use a small amount of location annotations to simultaneously performs disease identification and localization.

3. Method

We first briefly introduces our semi-supervised object detection learning formulation and training-mining framework. We present the proposed NOise Tolerant Ensemble R-CNN (NOTE-RCNN) detector.

3.1. Problem Formulation

We aim to train object detectors for target categories. For target categories, we have a small amount of seed bounding box annotations $\mathbf{B^0}$, as well as a large amount of image level annotations $\mathbf{A}$. We also have a pre-trained object detection $\mathcal{S}$ on source categories, which do not overlap with target categories.

3.2. Detector Training-Mining Framework

The object detectors are trained in a iterative training-mining framework, where the trained detector at iteration t is denoted as $\mathcal{T}^t$. The detector training-mining framework has the following steps.

Detector Initialization. A initial target detector $\mathcal{T}^0$ is initialized from the source detector $\mathcal{S}$ and trained using the seed bounding box annotations $\mathbf{B^0}$.

Box Mining. Box mining uses the the current detector $\mathcal{T}^{t-1}$ to mine a set of high quality bounding box annotation $\mathbf{B^t}$ for target categories from annotations with image-level labels $\mathbf{A}$. A bounding box is mined if it fulfills the following conditions: 1) its (predicted) label matches with the image-level groundtruth label; 2) the box's confidence score is the highest among all boxes with the same label; 3) its confidence score is higher than a threshold θ_b. The process can be summarized as $\mathbf{B^t} = \mathbf{M}(\mathbf{A}, \mathcal{T}^{t-1}, \theta_b)$, where $\mathbf{M}$ is the box mining function;

Detector Retraining. A new detector $\mathcal{T}^t$ is trained with the union of mined bounding boxes $\mathbf{B^t}$ and the seed bounding boxes $\mathbf{B^0}$. The parameters of the new detector $\mathcal{T}^t$ are initialized from the detector $\mathcal{T}^{t-1}$ from the previous iteration. The process can be summarized as $\mathcal{T}^t = \mathbf{R}(\mathbf{B^t}, \mathbf{B^0}, \mathcal{T}^{t-1})$, where $\mathbf{R}$ represents the re-training function.

3.3. NOTE-RCNN

NOTE-RCNN is designed to be tolerate to noisy box annotations that are generated in the training-mining framework.

There are three types of noise in the mined boxes: false negatives, false positives and box coordinate noise of the mined boxes. NOTE-RCNN is based on Faster RCNN, and it contains three improvements for noise tolerance: ensembling two classification heads and a distillation head to avoid overfitting on noisy labels and improve the mining precision, masking the negative sample loss in box predictor to get rid of the harm of false negative labels, and training box regression head only on seed annotations to eliminate the effect of inaccurate box coordinates of mined bounding boxes.

3.3.1 Recap of Faster RCNN architecture

In Faster RCNN, object locations are predicted in two stages: proposal prediction stage and box prediction stage. The first stage, called Region Proposal Network (RPN), outputs a set of class-agnostic proposal boxes for an image. It

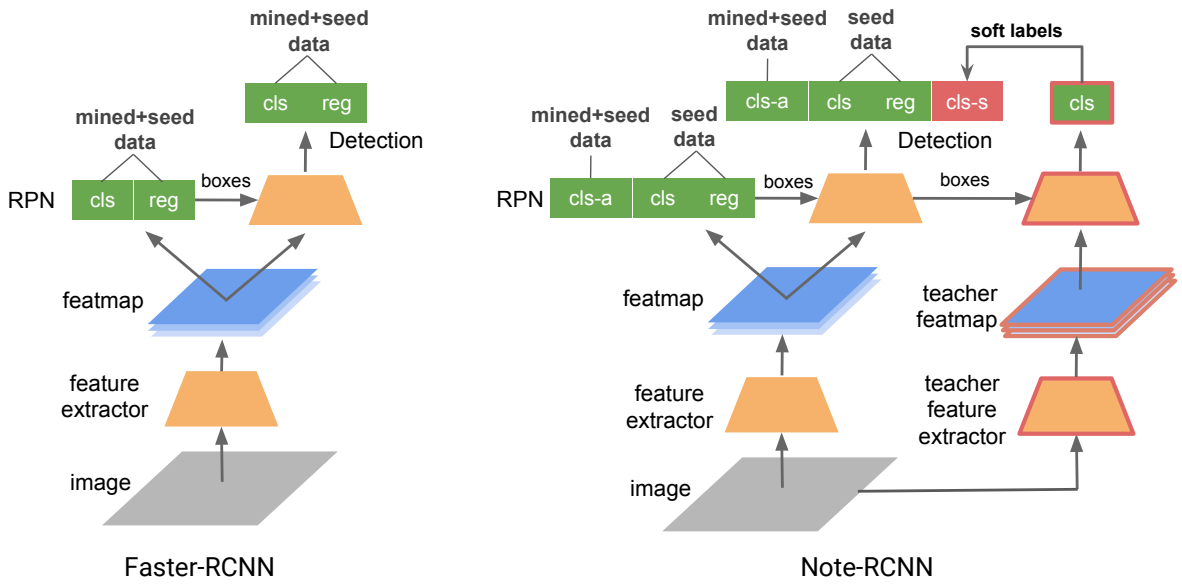

Figure 3. NOise Tolerant Ensemble R-CNN (NOTE-RCNN) architecture. There are three differences from standard Faster RCNN: additional classification heads in RPN and box predictor; noise tolerant training strategy; knowledge distillation from pre-trained detectors.

uses a feature extractor (*e.g.*, VGG-16, ResNet-101) to extract feature maps from an image, and it predicts proposal boxes using ROI pooled features in a set of predefined anchors in this feature map. We denote its classification head as **rpn-cls**, the box coordinate regression head as **rpn-reg**. An outline of the architecture is shown in the left part of Figure 3. The loss function of RPN is as follows:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (1)$$

where i is the index of an anchor and p_i is the predicted object probability of anchor i. The ground-truth label p_i^* is 1 if the anchor's overlap with groundtruth bounding box is larger than a threshold, and is 0 otherwise. t_i is a vector encoding of the coordinates the bounding box, and t_i^* is that of the ground-truth box associated with a positive anchor, $L_{cls} = -p_i^* log(p_i)$ is binary cross-entropy loss, L_{reg} is smooth L1 loss.

In the second stage, which is called box predictor network, features are cropped from the same intermediate feature maps for each proposal box, and they are resized to a fixed size. These features are fed to the box predictor network to predict class probabilities and class-specific box refinement for each proposal. We denote the classification head as **det-cls**, the boundary regression head as **det-reg**. The loss function for the second stage is similar to Equation 1. The only difference is that p_i is replaced by p_i^u, which is the predicted probability of category u; correspondingly, p_i^{u*} is 1 if the proposal box's overlap with a box with category u is larger than a threshold.

3.3.2 Box Predictor in NOTE-RCNN

In order to improve the noise tolerance, we use an ensemble of two classification heads in box predictor network (*i.e.* second stage of the detector). The seed classification head **det-cls** is trained only one seed bounding box annotations $\mathbf{B^0}$ so that it is not disturbed by the false negatives and false positives in the mined annotations $\mathbf{B^t}$. The mixed classification head **det-cls-a** utilizes both seed box annotations $\mathbf{B^0}$ and mined box annotations $\mathbf{B^t}$ for training. The consensus of the seed and mixed classification head is employed for a robust estimation of classification confidence.

The regression head **det-reg** is trained only one seed bounding box annotations $\mathbf{B^0}$, too, so that it is not affected by the inaccurate box coordinates in $\mathbf{B^t}$.

Filtering background proposal loss. Given that false negative is extremely hard to eliminate in mined bounding boxes, the losses of "background" proposals in **det-cls-a** are not used in loss to remove the effect of false negatives.

Specifically, if an image i is from mined box annotation set $\mathbf{B^t}$, then we mask the losses from the proposals that belong to "background" category (typical implementation uses index 0 for background); if the image is from seed box annotation set $\mathbf{B^0}$, then the loss is calculated normally. The

classification loss can be expressed as

$$L_{det-cls-a}(p_i, u, i) = -p_i^{u*} log(p_i^u) * \lambda(u, i),$$
$$\lambda(u, i) = \begin{cases} 0 & \text{u=0 \& } i \notin \mathbf{B^0} \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

During training, the loss function for the box predictor consists of the losses ofthree heads: **det-cls**, **det-reg** and **det-cls-a**, i.e., $L_{det} = L_{det-cls} + L_{det-cls-a} + L_{det-reg}$. During inference, the classification probability outputs from **det-cls** and **det-cls-a** are averaged.

3.3.3 RPN in NOTE-RCNN

Similarly, we add an additional binary classification head **rpn-cls-a** in RPN. Similar to box predictor, the seed classification head **rpn-cls** and the regression head **rpn-reg** are trained only on seed bounding box annotations $\mathbf{B^0}$. The mixed head **rpn-cls-a** uses both seed box annotations $\mathbf{B^0}$ and mined box annotations $\mathbf{B^t}$ for training. Different from box predictor, we don't zero the background loss if the training image is from mined annotation set, as RPN solves a binary classification problem and filtering background loss makes it unlearnable.

During training, the loss function for RPN comes from the three heads, **rpn-cls**, **rpn-reg** and **rpn-cls-a**, which can be expressed as $L_{rpn} = L_{rpn-cls} + L_{rpn-cls-a} + L_{rpn-reg}$. During inference, the classification probability outputs from **rpn-cls** and **rpn-cls-a** are averaged.

3.3.4 Knowledge Distillation as Supervision

We observe that although the source categories and target categories don't overlap with each other, they share some common visual characteristics, for example "bus" and "truck". To use those relationships, one can try to build a probability transformation matrix from the source categories to target categories using visual or semantic similarities among the categories (*e.g.* if number of source categories is m, n for target categories, then the transformation matrix would be $m * n$). We find that such direct transformation usually involves many noisy correlations, which will harm the final detection performance. Our motivation is not to directly affect the classification head of target categories, but to adjust the base feature extractor, and prevent it from overfitting on the noisy annotations. As the source detectors are trained on clean annotations, we expect to use probability distribution generated from source detectors as additional supervision to regularize the feature extractor in target detectors.

We added a knowledge distillation head **det-cls-s** to source detector S for additional noise tolerance, because it stops the target detector from overfitting to noisy annotations. During training, for a image I_k, we first forward I_k in the target detectors $\mathbf{T^t}$ to generate proposal boxes $\{P_k^t\}$. Then we forward the image I_k together with the proposals P_k^t to the source detector S to get the probability distribution on the source classes for every proposal. We use such distribution as a supervision to train **det-cls-s**. This process is known as knowledge distillation [10]. The loss function can be expressed as

$$L_{dist} = \frac{1}{N_{dist}} \sum_s \sum_j -p_s^{j*} log(p_s^j) \quad (3)$$

where j is the class index, s is the proposal index, p_s^{j*} is the probability of proposal s for class j from source detectors, and p_s^j is that from target detectors. The gradients generated from **det-cls-s** don't affect the parameters of **det-cls-a**, **det-cls** and **det-reg**, but they affect the feature extractor parameters.

As the source detectors are trained on large scale clean annotations, we expect to use probability distribution generated from source detectors as additional supervision to regularize the feature extractor in target detectors. Our motivation is not to directly affect the classification head of target categories, but to prevent the feature extractor from overfitting the noisy annotations.

4. Evaluation

In this section, we first present the implementation details of the whole detection system. We then introduce the benchmark datasets for evaluation. Third, we introduce our evaluation metrics and ablation studies. Finally, we discuss the experimental results on MSCOCO and ILSVRC.

4.1. Implementation Details

We use Inception-Resnet-V2 [24] as the feature extractor of the detector for all the experiments in this paper. The Inception-Resnet-V2 feature extractor is initialized from the weights trained on ImageNet classification dataset [22]. All input images are resized to $600 * 1024$. In the first stage, 300 proposal boxes are selected. We use SGD with momentum with batch sizes of 1 and learning rate at 0.0003. The system is implemented using the Tensorflow Object Detection API [13]. In all the experiments except the OID one, we employ 8 iterations of training-mining process, because we find the performance generally satuates after 8 iterations. In each iteration, the model is trained for 20 epochs. The mining threshold θ_b is set to 0.99 if no other specification is given.

4.2. Datasets

MSCOCO 2017. MSCOCO [18] contains 80 categories, which is a superset of PASCAL VOC [7] categories. We split both training and validation data to VOC categories (*i.e.* source categories) and non-VOC categories (*i.e.* target categories). If an image has both source category and target

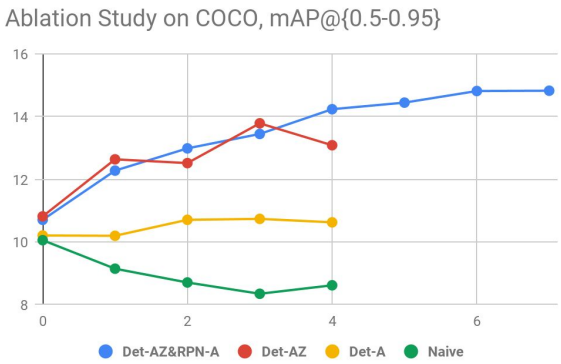

Figure 4. Ablation studies on MSCOCO 2017 dataset.

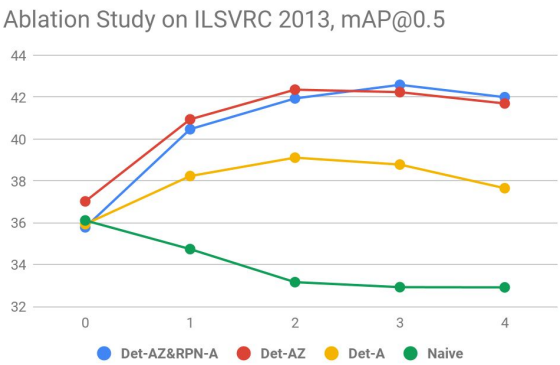

Figure 5. Ablation studies on ILSVRC 2013 dataset.

category bounding boxes, this image is used in both source category data and target category data, but source category and target category data only contains bounding boxes with the corresponding categories.

The source category training data is used to train source detectors. For target category training data, we randomly pick certain amount of images for each category as seed groundtruth bounding box annotations, and keep only image-level labels for the rest of images. We evaluate the target detectors on the target category validation data. To evaluate the method under varied amounts of seed annotations, we experiment with seed annotations with different average sizes: [12, 33, 55, 76, 96]. The average size means the average number of annotated images per category.

ILSVRC 2013. We follow the same settings as [11, 26, 27] for direct comparisons on ILSVRC 2013 [22]. We split the ILSVRC 2013 validation set into two subsets val1 and val2, and augment val1 with images from the ILSVRC 2013 training set such that each class has 1000 annotated bounding-boxes in total [9]. Among the 200 object categories in ILSVRC 2013, we use the first 100 in alphabetical order as sources categories and rest as target categories. We use all images of the source categories in augmented val1 set as the source training set, and that of the target categories in val2 set as source validation set. For target training set, we randomly select 10-20 images for each target category in augment val1 set as seed groundtruth bounding boxes, and use the rest of images as image-level labels by removing the bounding box information. All images of the target categories in val2 set are used as target validation set.

4.3. Experiment Settings

Evaluation metric. For object detection performance, we use the mean Average Precision (mAP), which is averaged mAP over IOU thresholds in [0.5 : 0.05 : 0.95]. We also report mAP@IOU 0.5. To measure the quality of mined box annotations, we report *box recall* and *box precision*. Box recall means the percentage of the true positive boxes in the mined annotations over all groundtruth boxes, Box precision means the percentage of the true positive boxes over all boxes in the mined annotations.

Ablation studies. To evaluate the contribution of each component in NOTE-RCNN, we design the following system variants for ablation studies:

(1) **Naive**: no additional classification head is added to RPN nor box predictor, *i.e*. stardard Faster RCNN; both mined annotation and seed groundtruth annotation are used to train the classification heads and regression heads (for both detection and RPN). (2) **Det-A**: we add the additional classification head **det-cls-a** to box predictor, but not to RPN; the original head **det-cls** and **det-reg** are trained by the seed groundtruth annotations, **det-cls-a** is trained on both seed groundtruth data and seed annotation; we don't zero the background sample loss in this variant. (3) **Det-AZ**: Similar to Det-A, but we zero the background sample loss in this variant. (4) **Det-AZ&RPN-A**: we add the additional classification heads to both RPN and detection part. **det-cls**, **det-reg**, **rpn-cls**, **rpn-reg** are trained on the seed groundtruth annotations, **det-cls-a** and **rpn-cls-a** are trained on both seed annotations and mined annotations; We zero the background sample loss on **det-cls-a**, but not on **rpn-cls-a**. (5) **Det-AZ&RPN-A&Distill**: Similar to Det-AZ&RPN-A, but we ad the distillation head.

4.4. Experiments and Discussions

Evaluation on additional classification heads. To show the contribution of each component, we do ablation studies on the additional heads and the system variants. Experimental results on MSCOCO are shown in Figure 4. For Naive, Det-A and Det-AZ, we stop training in 4 iterations, as the performance already decreases in iteration 4. For Det-AZ&RPN-A, we train it for 8 iterations. Experimental results on ILSVRC 2013 are shown in Figure 5. On this dataset, We train all system variants for 4 iterations. Iteration 0 means that the detector is only trained on the seed groundtruth box annotations. The performances on iteration

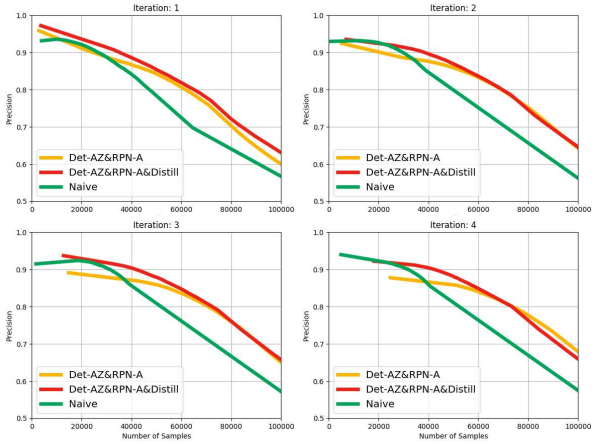

Figure 6. Comparison on "Box Precision vs Number of Samples Curve" of Mined Annotations on MSCOCO 2017.

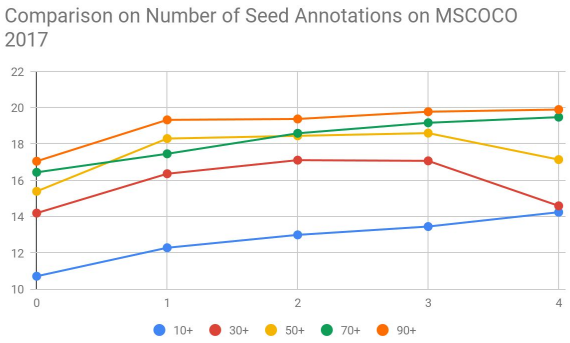

Figure 7. Comparison on different amount of seed annotations on MSCOCO 2017.

	Det-AZ&RPN-A		Det-AZ&RPN-A&Distill	
# iter	# boxes	prec(%)	# boxes	prec(%)
1	21542	90.0	22972	88.3
2	38323	87.1	32698	90.8
3	44223	86.6	38727	89.9
4	54680	84.9	41576	90.0
5	60899	83.7	42756	89.9

Table 1. Comparison between "with distillation" and "without distillation" on annotation mining on MSCOCO 2017, threshold θ_b is set to be 0.99.

0 for different system variants is slightly different, because we initialize each detector independently.

From Naive models, we can see that if we don't separate the seed groundtruth annotations with mined annotations, and just train the regression head and classification head with all data, the performance drops immediately after we add the mined data (iteration 1 and after). For Det-AZ and Det-A, it can be observed that zeroing the background loss gives significant improvements on both MSCOCO (in Figure 4) and ILSVRC (in Figure 5). Comparing Det-AZ&RPN-A and Det-AZ in MSCOCO (Figure 4), we can see that the performance of Det-AZ&RPN-A consistently increases, but that of Det-AZ starts to decrease after the 3rd iteration. We believe that the reason is that more accurate RPN and detection helps to improve the performance of each other. Specifically, the ensemble classification heads in RPN improve the proposal quality, resulting in the discovery of more object proposals; higher quality object proposals are beneficial to the detection performance; better detection performance leads to higher quality mined annotations, which in turn improves the RPN accuracy. Thus, applying ensemble classification heads to both RPN and box predictor are important for consistent performance increase. The difference between Det-AZ&RPN-A and Det-AZ on ILSVRC (in Figure 5) is not significant. The reason is that ILSVRC 2013 is a relatively simple dataset for detection, where an image usually only contains 1 to 2 objects of interest and the area of object is usually large, leading to lower mining annotation difficulty.

Using different amount of seed annotations. To evaluate the performance of the proposed method using different amount of seed bounding box annotations, we test NOTE-RCNN with varied sizes of seed annotation set on MSCOCO. The average sizes (*i.e.* average number of annotated images per category) tested are [12, 33, 55, 76, 96]. The method used for evaluation is Det-AZ&RPN-A. In Figure 7, we can see that the performance first increases and then goes down after saturation. The reason is that more and more annotations are mined in the mining iterations, but the quality (label precision) of the mined annotations gradually goes down (more discussion about mining quality is in next paragraph). On the one hand, more annotations help to improve the performance, on the other hand, worse quality annotations harm the performance. In general, more seed boxes used, more robust the mining would be. Overall, we can see that NOTE-RCNN provides steady performance improvements for all experiments, indicating the effectiveness of the proposed method when different amount of seed annotated images are used.

Bounding box mining quality. We evaluate the bounding box mining precision for Naive, Det-AZ&RPN-A and Det-AZ&RPN-A&Distill methods. First, we draw "box precision vs number of samples" curves of mined annotations on MSCOCO, shown in Figure 6. This curve is generated by varying the mining threshold θ_b from 0 to 1.0, and we show the part of curve that falls in between $[0, 10^5]$ samples. The results of 1st to 4th iterations are shown. We can see that the precision of Naive drops very fast when the number of samples increase; Det-AZ&RPN-A performs better than Naive when the number of samples is large; Det-AZ&RPN-A&Distill achieves the best precision performance.

We further compare the actual precision and number

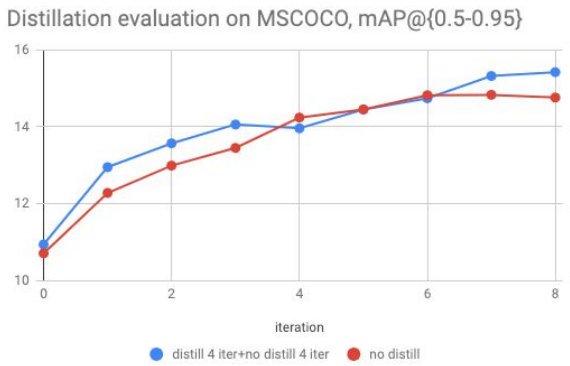

Figure 8. Comparison between "half-distill" and "no-distill" on target detector performance on MSCOCO 2017.

of boxes in each iteration between Det-AZ&RPN-A and Det-AZ&RPN-A&Distill by setting the θ_b as 0.99. As shown in Table 1, we can see that: (1) without using distillation, the precision decreases gradually, from 90.0% to 83.7%, with distillation, the precision is preserved at around 90.0%; (2) the increasing speed of mined box number of Det-AZ&RPN-A is higher than that of Det-AZ&RPN-A&Distill. Generally, it can be seen that Det-AZ&RPN-A performs better than Naive, which shows the effectiveness of the ensemble classification heads, and using distillation head further improves the mining precision by preventing the network from overfitting noisy labels.

Combining distillation in training-mining process. We find that the quantity (*i.e.* # boxes) and quality (*i.e.* box precision) of annotations are the two key factors that influence the detector performances: both higher quality and higher quantity result in better detectors. This inspires us to combine the distillation (higher quality) with non-distillation (larger quantity) method, called half-distill. We apply Det-AZ&RPN-A&Distill in the first 4 iterations and Det-AZ&RPN-A in the later 4 iterations. The experimental results are shown in Figure 8. We can see that: 1) in the beginning stage (first three iterations), the performance of "half-distill" is significantly better than that of "no-distill", because "half-distill" could generate higher quality of annotations; 2) in the middle stage (around 4 iterations), "no-distill" catches "half-distill", as "half-distill" suffers from fewer mined annotations; 3) in the final stage, after we switch the "half-distill" to "no-distill", the performance improves again. Note that the full supervision model with the same backbone net (inception-resnet) can achieve 37% average mAP on COCO dataset, there is still a big performance gap between fully supervised models and semi-supervised models.

Comparison with state-of-the-art methods. The most related work is SemiMIL [27], but it doesn't use seed box annotations for the target categories. For a fair comparison, we build two stronger baseline methods based on SemiMIL [27]. 1) SemiMIL+Seed+FRCN: We use SemiMIL to mine the box annotations from images, and then add the same seed annotations to the training set, following [27] to train a standard Faster RCNN. 2) SemiMIL+Seed+NOTE-RCNN: Similar to the previous baseline, but we replace the standard Faster RCNN by NOTE-RCNN.

model	backbone	mAP
LSDA [11]	alexnet	18.1
Tang *et al*. [26]	alexnet	20.0
FRCN+SemiMIL [27]	alexnet	23.3
FRCN+SemiMIL [27]	inception-resnet	36.9
FRCN+SemiMIL+Seed	inception-resnet	38.7
NOTE-RCNN+SemiMIL+Seed	inception-resnet	39.9
Ours(wo/ distill)	inception-resnet	42.6
Ours(w/distill)	inception-resnet	43.7

Table 2. Comparison with state-of-the-art on ILSVRC 2013

The performance of state-of-the-art methods and the new baselines are shown in Table 2. Comparing FRCN+SemiMIL+Seed and FRCN+SemiMIL, we can see that by adding seed annotations, the performance increases by 1.8%. By changing Faster RCNN to NOTE-RCNN, the performance increases by 1.2%, which shows the effectiveness of NOTE-RCNN in handling the noisy annotations. Our method (wo/ distill) achieves 42.6% mAP and outperforms all state-of-the-art methods; by applying distillation (w/ distill), we further improve the performance to 43.7%, which is the best among all methods.

5. Conclusion and Future Work

We proposed a new semi-supervised object detection formulation, which uses large number of image levels labels and a few seed box level annotations to train object detectors. To handle the label noises introduced in training-mining process, we proposed a NOTE-RCNN object detector architecture, which has three highlights: an ensemble of two classification heads and a distillation head to improve the mining precision, masking the negative sample loss in box predictor to avoid the harm of false negatives, and training box regression heads only on seed annotations to eliminate the harm from inaccurate box boundaries. Evaluations were done on ILSVRC and MSCOCO dataset, we showed the effectiveness of the proposed methods and achieved the state-of-the-art performance. In the future, we plan to add human annotation to the training-mining iterations. We believe a combination of human annotation and accuracy box mining can further improve the detector performance.

References

[1] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2846–2854, 2016. 3

[2] Jacob Buckman, Danijar Hafner, George Tucker, Eugene Brevdo, and Honglak Lee. Sample-efficient reinforcement learning with stochastic ensemble value expansion. *arXiv preprint arXiv:1807.01675*, 2018. 2

[3] Hyunsun Choi and Eric Jang. Generative ensembles for robust anomaly detection. *arXiv preprint arXiv:1810.01392*, 2018. 2

[4] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Weakly supervised object localization with multi-fold multiple instance learning. *IEEE transactions on pattern analysis and machine intelligence*, 39(1):189–203, 2017. 1, 2, 3

[5] Thomas Deselaers, Bogdan Alexe, and Vittorio Ferrari. Weakly supervised localization and learning with generic knowledge. *International journal of computer vision*, 100(3):275–293, 2012. 3

[6] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1-2):31–71, 1997. 1, 2

[7] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015. 5

[8] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 3

[9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 3, 6

[10] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2, 5

[11] Judy Hoffman, Sergio Guadarrama, Eric S Tzeng, Ronghang Hu, Jeff Donahue, Ross Girshick, Trevor Darrell, and Kate Saenko. Lsda: Large scale detection through adaptation. In *Advances in Neural Information Processing Systems*, pages 3536–3544, 2014. 2, 3, 6, 8

[12] Ronghang Hu, Piotr Dollr, Kaiming He, Trevor Darrell, and Ross Girshick. Learning to segment every thing. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3

[13] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7310–7311, 2017. 5

[14] Zequn Jie, Yunchao Wei, Xiaojie Jin, Jiashi Feng, and Wei Liu. Deep self-taught learning for weakly supervised object localization. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 4294–4302. IEEE, 2017. 3

[15] Vadim Kantorov, Maxime Oquab, Minsu Cho, and Ivan Laptev. Contextlocnet: Context-aware deep network models for weakly supervised localization. In *European Conference on Computer Vision*, pages 350–365. Springer, 2016. 3

[16] Zhe Li, Chong Wang, Mei Han, Yuan Xue, Wei Wei, Li-Jia Li, and Li Fei-Fei. Thoracic disease identification and localization with limited supervision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3

[17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007. IEEE, 2017. 1

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2, 5

[19] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 1

[20] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1

[22] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 2, 5, 6

[23] Hyun Oh Song, Ross Girshick, Stefanie Jegelka, Julien Mairal, Zaid Harchaoui, and Trevor Darrell. On learning to localize objects with minimal supervision. In *International Conference on Machine Learning*, pages 1611–1619, 2014. 3

[24] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. 2017. 5

[25] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. Multiple instance detection network with online instance classifier refinement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2843–2851, 2017. 1, 2, 3

[26] Yuxing Tang, Josiah Wang, Boyang Gao, Emmanuel Dellandréa, Robert Gaizauskas, and Liming Chen. Large scale semi-supervised object detection using visual and semantic knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2119–2128, 2016. 2, 3, 6, 8

[27] Jasper Uijlings, Stefan Popov, and Vittorio Ferrari. Revisiting knowledge transfer for training object class detectors. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 2, 3, 6, 8

[28] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013. 2

[29] Xiaopeng Zhang, Jiashi Feng, Hongkai Xiong, and Qi Tian. Zigzag learning for weakly supervised object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 2, 3

[30] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European conference on computer vision*, pages 391–405. Springer, 2014. 2, 3