Supplementary Material: Noisy Annotation Refinement for Object Detection

Jiafeng Mao
mao@hal.t.u-tokyo.ac.jp
Qing Yu
yu@hal.t.u-tokyo.ac.jp
Yoko Yamakata
yamakata@hal.t.u-tokyo.ac.jp
Kiyoharu Aizawa
aizawa@hal.t.u-tokyo.ac.jp

The University of Tokyo
Tokyo, Japan

1 Definition of noise

1.1 Symmetry Label Noise

The definition of transition matrix $\mathcal{Q}$ of symmetric noise is as follow,

$$\mathcal{Q} = \begin{bmatrix} 1-r & \frac{r}{n-1} & \cdots & \frac{r}{n-1} & \frac{r}{n-1} \\ \frac{r}{n-1} & 1-r & \frac{r}{n-1} & \cdots & \frac{r}{n-1} \\ \vdots & & \ddots & & \vdots \\ \frac{r}{n-1} & \cdots & \frac{r}{n-1} & 1-r & \frac{r}{n-1} \\ \frac{r}{n-1} & \frac{r}{n-1} & \cdots & \frac{r}{n-1} & 1-r \end{bmatrix} \tag{1}$$

where r is the noise rate and n is number of the class.

1.2 Pair Label Noise

The definition of transition matrix $\mathcal{Q}$ of pair noise is as follow,

$$\mathcal{Q} = \begin{bmatrix} 1-r & r & 0 & \cdots & 0 \\ 0 & 1-r & r & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & & 1-r & r \\ r & 0 & \cdots & 0 & 1-r \end{bmatrix} \tag{2}$$

where r is the noise rate.

1.3 Uniform Localization Noise

If the original bounding boxes are presented as $[x_1, y_1, x_2, y_2]$, the noisy bounding boxes $[x'_1, y'_1, x'_2, y'_2]$ can be formulated as follows:

$$\begin{cases} x'_1 = x_1 + \delta_1(x_2 - x_1), \\ x'_2 = x_2 + \delta_2(x_2 - x_1), \\ y'_1 = y_1 + \delta_3(y_2 - y_1), \\ y'_2 = y_2 + \delta_4(y_2 - y_1), \end{cases} \tag{3}$$

where $\delta_i \sim U(-N_{BBox}, N_{BBox})$.

1.4 Gaussian Localization Noise

We use the same way as above to create the Gaussian noise dataset. The difference is that the $\delta_i \sim N(0, \sigma^2)$ of Gaussian noise datasets follow the Gaussian distribution.

2 Implementation Details

Although our explanation about the proposal assumes that there is only one object in each image and the training batch size is 1, our experiments train the Faster R-CNN with batch size as 2. In practice, we apply the process we described in Section 3 to each image and each object. During the pre-forward stage, each annotated object will generate one corresponding classification loss. We record the value of the loss and then judge their correctness and refine them separately, as shown in Algorithm. 1.

3 Analysis on Hyper-parameters

There are three hyperparameters need to be determined for our proposal. They are the weight α for the initial bounding box correction in Eq. (19), length N for queue $\mathcal{Q}$, and the acceptance rate r for CINJ.

The acceptance rate r should be close to the percentage of correct class labels in the training dataset, which can be estimated by using a small amount of training sample. However, when the dataset is extremely noisy and the dataset is unbalance, setting an acceptance rate close to the percentage of correct class labels may prevent the detector from learning hard classes. In such case, slightly increase the acceptance rate can allow the detectors to learn from hard classes. In our experiments, the acceptance rate r for datasets with N_{label} as $20\%, 40\%$ and 60% are set as $80\%, 60\%$ and 50%.

The quantitatively sensitive analysis on N is shown on Table. 1. The performance of our proposal is not sensitive to N, all of experiments with N larger than 128 achieve relatively close results. α is the only tunable parameter of our proposal. The quantitatively sensitive Analysis on α is shown on Table. 2. Tuning α from 0.2 to 0.4 can achieve an optimal performance, and all setting of α from 0.1 to 0.5 significantly outperform the baseline.

Algorithm 1: Overall Architecture

Input: Image $\mathcal{X}$, noisy annotation $\{\mathcal{Y},\mathcal{B}\}$.;
$\mathcal{Q} \leftarrow \{\infty\}^N$;
while *not MaxIters* **do**
  for $x \in Batch$ **do**
    Image x with annotation $Y = \{y_i\}, B = \{b_i\}$;
    Generate proposals P by RPN(x);
    for $\{y,b\} \in \{Y,B\}$ **do**
      $b^*, P_b \leftarrow$ Center-Matching(b,P);
      for $b_i \in \{b^*\} \cup P_b$ **do**
        Calculate score $p(x,b_i)$ by Pre-Forward;
        $b_i^r \leftarrow$ Regress(b_i) by Pre-Forward;
        $B_r \leftarrow B_r \cup \{b_i^r\}$;
      end
      Calculate b_m^* by Eq. (14);
      Calculate classification loss $\mathcal{L}(b^*,y)$;
      $clean(y) \leftarrow$ CINJ$(\mathcal{L}(b^*),\mathcal{Q})$;
      if $clean(y)$ **then**
        $Y^* \leftarrow Y^* \cup \{y\}, B^* \leftarrow B^* \cup \{b_m^*\}$;
      else
        Pseudo-label $y^* \leftarrow \arg\max_c p(c|x,b^*)$;
        if $p(y^*|x,b^*) > \sum_{c \neq y^*} p(c|x,b^*)$ **then**
          $Y^* \leftarrow Y^* \cup \{y^*\}, B^* \leftarrow B^* \cup \{b_m^*\}$;
        end
      end
      $\mathcal{Q} \leftarrow \mathcal{Q} + \{\mathcal{L}(b^*)\}$;
      $\mathcal{Q} \leftarrow \mathcal{Q} - \mathcal{Q}_0$;
    end
    Train model with updated annotation $\{Y^*,B^*\}$;
  end
end

4 Analysis on Thresholds

There are 2 thresholds need to be determined for our proposal. They are the threshold $\mathcal{T}_{CM}$ of matching region proposals to annotated bounding boxes in Eq.(18), and the label refinement threshold $\mathcal{T}_{refine}$ in Eq.(13). The quantitatively sensitive analysis on $\mathcal{T}_{CM}$ and $\mathcal{T}_{refine}$ are shown on Table. 3 and Table. 4, respectively. The performance of our proposal is not sensitive to both of these two thresholds, all of the experiments achieve relatively close results.

5 Evaluation on Refined Datasets

Our proposal alternately update the noisy annotation and the parameters of the detector. To prove the effectiveness of our noisy annotation refinement, we record the refined annotations in last training epoch and evaluate them by clean annotations. Table. 5 shows the results of

N_{Label}	N				Baseline
	64	128	258	512	
20%	79.6	**79.9**	79.7	79.8	75.2
40%	78.8	79.3	**79.6**	79.3	71.3
60%	76.3	**76.6**	76.3	76.4	66.8

Table 1: mAP of Faster R-CNN trained on datasets with classification label noise by different N.

N_{BBox}	α					Baseline
	0.1	0.2	0.3	0.4	0.5	
20%	78.0	**78.4**	77.9	77.4	77.8	75.6
40%	72.3	73.2	**73.4**	73.1	72.8	58.9

Table 2: mAP of Faster R-CNN trained on datasets with localization annotation noise by different α.

N_{Label}	$\mathcal{T}_{CM}$				Baseline
	0.8	0.85	0.9	0.95	
20%	79.6	**79.9**	79.7	79.8	75.2
40%	78.8	79.3	**79.6**	79.3	71.3
60%	76.3	**76.6**	76.3	76.4	66.8

Table 3: mAP of Faster R-CNN trained on datasets with classification label noise by different $\mathcal{T}_{CM}$.

N_{BBox}	$\mathcal{T}_{refine}$				Baseline
	0.4	0.5	0.6	0.7	
20%	78.0	**78.4**	77.9	77.8	75.6
40%	72.3	73.2	**73.4**	72.8	58.9

Table 4: mAP of Faster R-CNN trained on datasets with localization annotation noise by different $\mathcal{T}_{refine}$.

N_{BBox}	20%				40%			
N_{Label}	0%	20%	40%	60%	0%	20%	40%	60%
CorLoc$_{noisy}$	54.67	44.93	45.10	45.52	9.02	8.66	8.73	8.79
CorLoc$_{cm}$	57.35	57.23	56.97	57.62	15.44	15.39	15.15	15.38
CorLoc$_{final}$	**77.21**	**75.54**	**74.68**	**73.86**	**38.08**	**37.68**	**34.13**	**34.17**

Table 5: Correct localization rate(%) of refined noisy annotations.

N_{BBox}	0%			20%			40%		
N_{Label}	20%	40%	60%	20%	40%	60%	20%	40%	60%
TP	93.56	95.68	83.27	92.72	95.63	83.05	91.03	94.55	82.11
TN	97.07	94.44	98.6	96.79	94.77	98.23	96.65	93.94	97.06
FP	2.93	5.56	1.40	3.21	5.23	1.77	3.35	6.06	2.94
FN	6.44	4.32	16.73	7.28	4.37	16.95	8.97	5.45	17.89
N^*_{Label}	**3.48**	**6.31**	**13.97**	**4.23**	**6.49**	**13.81**	**5.36**	**8.45**	**16.21**

Table 6: Accuracy(%) of the class label noise refinement. TP denotes the percentage of noisy labels detected to be noisy, and TN denotes the percentage of clean labels judged to be clean. FP and FN denote the percentage of clean labels judged to be noisy and noisy labels judged to be clean, respectively. N^*_{Label} denotes the noise rate of classification labels after the judgment and refinement.

the localization annotation refinement. We compare the refined localization annotation with clean dataset. When a refined bounding box has an IOU with any bounding box in clean annotation larger than 0.7, it is considered as correct localization. The first row shows Corloc of the noisy annotation, second row and last row show the CorLoc after center-matching and final correction, respectively. Table. 6 shows the results of the classification label judgment and refinement. Although severe localization annotation noise and the classification label noise are entangled together, our proposal can still significantly reduce the class label noise rate.

6 Loss Distribution of Pair Label Noise

Although the pair noise is much more challenging than symmetry label noise, the training losses for noisy and clean annotations are still clearly separated, as shown in Fig. 1. This fact ensure the effectiveness of our method on more challenging noise models such as pair noise.

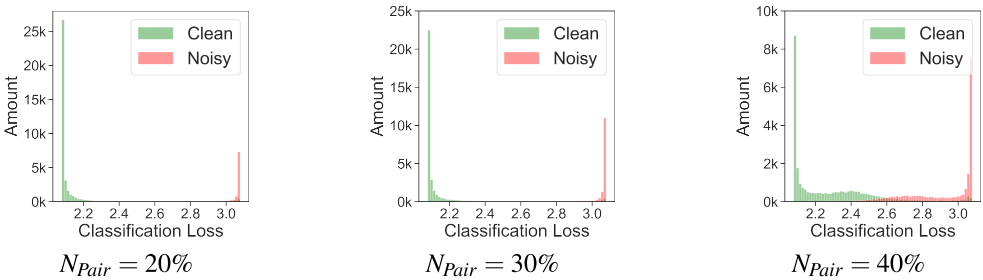

Figure 1: Distribution of the classification loss of datasets with pair noise.