PAPER

Noisy Localization Annotation Refinement for Object Detection*

Jiafeng MAO^{†a)}, Qing YU[†], Nonmembers, and Kiyoharu AIZAWA[†], Fellow

SUMMARY Well annotated dataset is crucial to the training of object detectors. However, the production of finely annotated datasets for object detection tasks is extremely labor-intensive, therefore, cloud sourcing is often used to create datasets, which leads to these datasets tending to contain incorrect annotations such as inaccurate localization bounding boxes. In this study, we highlight a problem of object detection with noisy bounding box annotations and show that these noisy annotations are harmful to the performance of deep neural networks. To solve this problem, we further propose a framework to allow the network to modify the noisy datasets by alternating refinement. The experimental results demonstrate that our proposed framework can significantly alleviate the influences of noise on model performance.

key words: object detection, robust learning, noisy localization annotation, joint optimization

1. Introduction

In recent years, with the development of deep neural networks (DNNs), supervised object detection models have achieved increasingly effective results. The representative ones are the R-CNN series [2]–[5], as the two-stage models and the YOLO series [6]–[9], as well as the SSD [10] for single-stage models. However, these methods need to be trained on datasets containing precise annotation, such as MSCOCO [11] and ImageNet [12], which are expensive and time-consuming to produce. At the same time, some attempts have been made to achieve more economical and cheaper dataset annotations. Some studies [13], [14] aim to reduce the labor cost of data annotation, and some others [15] try to collect data from the website. However, these datasets are not guaranteed to be completely accurate. Even in the mainstream datasets, which are well labeled by humans, ambiguities and uncertainties are still introduced [16] as noise due to cropping and occlusion of the objects. Owing to the interference of noise, the models trained on such datasets tend to perform worse than models trained on precise datasets. If we can train a model that is robust to noise on a sub-precise dataset, it will be possible to reduce the labor cost of data annotation.

Training on unreliable datasets is also inevitable for some tasks such as weakly-supervised learning [17] and semi-supervised learning [18]. In these tasks, it is common to use the existing model to generate relatively low-quality labels on the unlabeled dataset and then retrain the model on these generated annotations. These relatively low-quality labels can also be regarded as noisy. Therefore, the study about noise-supervised object detection is not only helpful to reduce the labor cost of data annotations, but also crucial to downstream tasks.

There could be several types of noise existing in a noisy object detection dataset, and these noises will have different influences to the model, which make the noisy problem much more challenging. This study focuses on training object detectors on datasets with noisy localization annotations (bounding boxes).

First, we train Faster R-CNN [2] and as the baseline to evaluate the impact of different noise levels on the object detector by manually introducing varying degrees of random localization noise into the training split of PASCAL VOC2007 [19]. All trained models are tested on the clean test split of PASCAL VOC2007 and the results show that the higher the noise level is, the worse is the model performance. To address this problem, we further propose a framework, named annotation refinement, to allow the network to refine noisy bounding box annotations by joint optimization as shown in Fig. 1, which significantly reduces the negative effects of noisy annotations on model performance.

Our proposed framework is generic and it is applicable to other object detection methods. We conduct experiments on RefineDet [20], one of the latest single-stage object detectors, to demonstrate the generality of our proposal.

Furthermore, we also apply our method to a real-world dataset. The experimental results show that our method can effectively correct the localization annotations with notice-

Fig. 1 Concept of the proposed framework. The network parameters and annotations are alternatively updated for each epoch.

Manuscript received January 31, 2021.

Manuscript revised May 4, 2021.

Manuscript publicized May 25, 2021.

[†]The authors are with the Dept. of Information and Communication Eng., The University of Tokyo, Tokyo, 113–0033 Japan.

*A preliminary version of this paper was presented at ICIP 2020 as “Noisy Localization Annotation Refinement for Object Detection” [1] by the same author. Additional experiments and visualizations have been added in this version.

a) E-mail: mao@hal.t.u-tokyo.ac.jp

DOI: 10.1587/transinf.2021EDP7026

able deviations in the real-world dataset.

The main contributions of this paper is summarized as follows:

- We demonstrate the negative effects of noisy localization annotations on Faster R-CNN by adding artificial noise with different noise levels.
- We propose a robust training strategy with joint optimization to reduce the impact of noise on the training process by modifying the noisy annotations in the training dataset.
- We demonstrate the generality of our proposal by additional experiments on RefineDet and conduct the quantitative sensitive analysis on hyperparameters.
- We evaluate our method across not only artificial datasets but also real-world datasets.

2. Related Works

Significant progress has been made for training classifier with noisy class labels on the general image classification task. Some methods utilized several regularization techniques, including weight decay, dropout [21], and adversarial training [22], to prevent DNNs from over-fitting noisy data. Some studies [23]–[25] used the noise transition matrix to modify the cross-entropy loss for classification. Zhang et al. [26] trained DNNs on mix-up data created by linear combination of labels and images to improve the generalization ability of the DNNs. Tanaka et al. [27] trained classifiers on noisy datasets with different noise rates and observed that the training loss increases with an increase in noise level. They also proposed a joint optimization framework to correct labels during training through replacing the labels by the outputs of the classifier to decrease the training loss.

By contrast, there are few studies focusing on noisy localization annotation. Chadwick et al. [28] utilized the co-teaching process for training on a noisy dataset, where the training of two networks is necessary. He et al. [16] considered the localization annotation in the standard dataset to be ambiguous and then proposed the KL-loss to deal with the annotation. Unlike these studies, we focus on recovering the accurate bounding boxes of the training data from the noisy annotations to improve the performance of the model in the object detection task.

3. Proposed Method

In this section, we introduce our framework for training detectors with noisy localization annotations. Our framework refines the annotations of bounding boxes for each epoch using the prediction of the network. First, we explain the problem statement in Sect. 3.1. Next, we describe our idea of annotation refinement in Sect. 3.2, followed by the refinement strategy in Sect. 3.3.

Fig. 2 Visualization of annotation in noisy datasets. r denotes the percentage of noisy bounding boxes, and range indicates the degree of inaccuracy by defining a uniform distribution, as Eq. (2).

3.1 Problem Statement

As we focus on training detectors with noisy localization annotations, we manually create several noisy datasets by introducing different levels of noise into clean training datasets. Only the coordinates of bounding boxes are changed, where the number of bounding boxes and class label of each bounding box remain unchanged. In this study, all mentioned annotations are assumed to be noisy unless otherwise noted.

At the t -th epoch of the training of a detector, the annotation involved and prediction of the detector for a single image i can be formed as sets $\mathbf{G}_i^{(t)}$ and $\mathbf{P}_i^{(t)}$, which include the M and N vectors (M bounding boxes in the annotation and N bounding boxes in the prediction), respectively. As the prediction is not guaranteed to be correct, M is not necessarily equal to N . Each vector includes 5 elements and the first 4 elements in each vector represent the coordinates of the bounding box as $[x_{min}, y_{min}, x_{max}, y_{max}]$. The last element represents the class label of each bounding box. In this work, we aim to refine the annotation $\mathbf{G}_i^{(t+1)}$ with $\mathbf{G}_i^{(t)}$ and $\mathbf{P}_i^{(t)}$ to train a robust detector for noisy annotations.

3.2 General Idea

To refine the noisy annotations in each epoch, we built a framework based on the joint optimization framework [27] that updates the annotations and network parameters alternately.

In the image classification task, Tanaka et al. [27] observed that the training loss increases with an increase in noise level, which means the noise of the dataset can be reduced by modifying the labels of the dataset towards the direction of decreasing the training loss. Based on their idea,

Fig. 3 Noisy annotation and the model's prediction of the same image. This is an artificial example to illustrate a typical situation during training.

we also observe the same relationship between training loss and noise level during the training of object detectors with noisy datasets, which will be detailed in Sect. 4.3. This observation shows that the general conception of optimizing noisy annotations towards the direction of decreasing the training loss can also be utilized in object detection tasks.

In the image classification task, the modification of the labels of the dataset to decrease the training loss can be achieved by replacing the label of a sample by the prediction of the network directly. However, in the object detection task, an image may contain multiple targets, which means multiple bounding boxes and class labels should be detected for each image. For example, in Fig. 3, the noisy annotation is partially accurate as it has the correct class labels, even if it is highly noisy. Conversely, detectors trained on noisy annotations tend to miss-predict objects and sometimes output multiple bounding boxes for a single object. In this situation, directly updating the annotations of the bounding boxes by the predictions of the detector destroys the annotations in the dataset instead of refining them. To prevent this problem, we propose the following refinement strategy.

3.3 Alternating Refinement

To prevent annotations being affected by worse predictions, we only update the annotation of the object when the object is detected by the detector, while the annotations of undetected objects remain unchanged. Furthermore, when a single object is detected as multiple bounding boxes, we only preserve the bounding box having the largest intersection over union (IoU) with the annotated bounding box to reduce the redundancy. These strategies can ensure that the refined annotations contain the same number of objects as the original annotations.

We first train the detector on the noisy dataset for several epochs by stochastic gradient descent (SGD) to give the model the essential ability to detect objects. Then we alternately optimize the model's parameters and refine each bounding box in annotations for each epoch.

Specifically, for each annotated bounding box g , we first find out all predicted bounding boxes that has the same label as the annotation and select the bounding box p^* having the largest IoU with the annotated bounding box g . If the IoU between the g and p^* is larger than a threshold, we update the annotated bounding box by the weighted average

Algorithm 1 Refining the annotation using the prediction

Input: $G_i^{(t)}$, which denotes a set of several bounding boxes with class labels of the annotation of the i -th image in the t -th epoch. $P_i^{(t)}$, which denotes the prediction of the detector and has a similar form to $G_i^{(t)}$.

Output: $G_i^{(t+1)}$, which denotes the refined annotations for training the detector in the next epoch.

```

1:  $G_i^{(t+1)} \leftarrow \phi$ 
2: for each bounding box  $g \in G_i^{(t)}$  do
3:    $list \leftarrow \{\}$ 
4:   for each bounding box  $p \in P_i^{(t)}$  do
5:     if  $Class(g) = Class(p)$  then
6:        $list \leftarrow list \cup \{p\}$ 
7:     end if
8:   end for
9:    $p^* \leftarrow \arg \max_p \{IoU(p, g) \mid p \in list\}$ 
10:  {Select the predicted bounding boxes having the same class with  $g$ .}
11:  {Select the bounding box having the largest IoU with the annotated bounding box.}
12:  if  $IoU(p^*, g) > threshold$  then
13:     $g^* \leftarrow update(p^*, g)$  as Eq. (1)
14:     $P_i \leftarrow P_i - \{p^*\}$ 
15:  else
16:     $g^* \leftarrow g$ 
17:  end if
18:  {Refine the annotated bounding box if the IoU is larger than the threshold.}
19:   $G_i^{(t+1)} \leftarrow G_i^{(t+1)} \cup \{g^*\}$ 
20: end for

```

as follow:

$$g^* = \frac{w \cdot p^* + g}{w + 1}, \quad (1)$$

where w denotes the weight of the predicted bounding box, and g and p^* denote the annotated bounding box and selected prediction box, respectively. Once a predicted bounding box is used for the updating, it is not used again in this iteration. The updating strategy is summarized as Algorithm 1.

3.4 Hyperparameters

There are two hyperparameters, IoU threshold and weight of the predicted bounding box w , need to be decided. The IoU threshold will affect the harshness of judging whether an object is detected. A small threshold means that even if the difference between the predicted bounding box and the annotated bounding box is large, they will still be judged as the same object. Therefore, when the noise level of a dataset is large, a small threshold will make the noisy annotations easier to be corrected, and a high threshold will prevent the annotations from being affected by wrong predicted bounding boxes. The value of the threshold should be negatively related to the noise level of the dataset. Specifically, modifying a extremely noisy dataset requires a small threshold, since the IoUs between good predictions and noisy annotations are small, while modifying a slightly noisy dataset requires a high threshold, since the IoUs between good predictions and noisy annotations are large. In most cases, simply

setting the IoU threshold to 0.5 can achieve a good performance.

The weight of the predicted bounding box w will affect the calculation of the Eq. (1). The larger value of w makes the predicted bounding boxes have a greater impact on the correction process. Since the refinement will be performed many times, the difference between the modified annotation and the prediction will become small as the training and refinement proceed, which makes w basically only affect the effect of the initial number of refinements, while the final correction result is insensitive to w . For simplicity, we set the $w = 1$, which make the Eq. (1) become a plain average, for all experiments in Sect. 4.3 and Sect. 4.4.

4. Experiment on Artificial Noisy Datasets

In this section, we create several artificial noisy datasets to observe how different levels of localization noise affect the detector.

4.1 Datasets

To simulate the dataset with localization noise, we manually introduce two kinds of noisy bounding boxes into clean PASCAL VOC2007 datasets.

4.1.1 Uniform Noise

First, we inject uniform noise $U(a, b)$ following the uniform distribution into the training split of PASCAL VOC2007 dataset. We set two parameters, the noise rate r and noise range $[a, b]$, to measure the level of noise.

The noise rate r determines the percentage of the bounding boxes in the annotations of the entire dataset, where the noisy annotations are chosen at random. If the bounding boxes are presented as $[x_1, y_1, x_2, y_2]$, we add a uniform noise to each coordinate of the chosen bounding boxes independently, as follows:

$$\begin{cases} x'_1 = \frac{1}{2}(x_1 + x_2) - \frac{\delta_1}{2}(x_2 - x_1), \\ x'_2 = \frac{1}{2}(x_1 + x_2) + \frac{\delta_2}{2}(x_2 - x_1), \\ y'_1 = \frac{1}{2}(y_1 + y_2) - \frac{\delta_3}{2}(y_2 - y_1), \\ y'_2 = \frac{1}{2}(y_1 + y_2) + \frac{\delta_4}{2}(y_2 - y_1), \end{cases} \quad (2)$$

where $\delta_i \sim U(a, b)$. We first choose r percent of the bounding boxes. It is noteworthy that all the class labels and number of bounding boxes in the training dataset remain unchanged. We set r as 60%, 80%, and 100%, $[a, b]$ as $[0.7, 1.3]$, $[0.5, 1.5]$, and $[0.3, 1.7]$, and obtain 9 noisy datasets with different levels of noise from their combination. Figure 2 shows the visualization of the datasets with different noise levels.

We train object detectors on these noisy datasets to evaluate the impact of different levels of noise on the model's performance. Subsequently, we apply our proposed

framework on these noisy datasets to verify the effectiveness of our method.

4.1.2 Gaussian Noise

We also introduce Gaussian noise, which is more common in real-world datasets, to simulate real-world noise. We use the same method as above to create the Gaussian noise dataset. The difference is that the $\delta_i \sim N(\mu, \sigma^2)$ of Gaussian noise datasets follow the Gaussian distribution. The noise rate and μ for each Gaussian noise dataset are set as 100% and 1, respectively. We set σ as 0.25, 0.30, and 0.35 to obtain 3 Gaussian noise datasets with different levels of noise.

4.2 Experiments on Faster R-CNN

Though the proposed method can be applied to any detection algorithm, we first fine-tuned Faster R-CNN, whose backbone is VGG16 pre-trained on Imagenet, as the baseline. We use a SGD optimizer to train the network for 14 epochs with a weight decay of 5×10^{-4} and learning rate of 10^{-3} . The learning rate decays to 10^{-4} at the end of the 10-th epoch.

When training the model with our proposal, we start annotation refinement from the 9-th epoch. As we mentioned in Sect. 3, we refine the annotations using current annotations and predictions for each image by Algorithm 1 and these refined annotations supervise the training of the next epoch. The hyperparameters w and t are set as 1 and 0.5 for all experiments, respectively.

4.3 Results

Figure 4 shows how the losses change when models are trained with different levels of localization noise. Figure 4(a) shows the training loss curves with a fixed noise rate, whereas Figure 4(b) shows the training loss curves with a fixed noise range. These experimental results indicate that a higher noise level (noise rate or noise range) leads to a higher training loss, which means that it is possible to reduce the noise with the aim of decreasing the loss. Table 1 shows the test performance on the clean test set of PASCAL VOC2007. Our method achieved a better performance than the baseline by mitigating the effects of noise on the detector effectively. Additionally, the margin between our method and the baseline is larger in situations where the noise is more noticeable.

Figure 5 shows the visualization of the noisy bounding boxes in original noisy annotations and the bounding boxes corrected by our method, when the model is trained on the noisy dataset with $r = 100\%$ and $range = [0.3, 1.7]$. The red bounding boxes are the noisy annotations that are used in training, and the green bounding boxes shows the annotations after modification. Figure 6 shows some imperfect correction cases. Limited by the performance of the model trained on the noisy dataset, not all corrections can get perfect results. However, the noise level of the data is

Fig. 4 Convergence of training losses during the training of the detector on noisy annotations.

Fig. 5 Visualization of the correction results of artificial-noised PASCAL VOC 2007 dataset with the noise parameters $r = 100\%$, $range = [0.3, 1.7]$. The red bounding boxes are the noisy annotations that are at the start of training, the green bounding boxes shows the annotations after modification.

Fig. 6 Examples of imperfect bounding box corrections with the noise parameters $r = 100\%$, $range = [0.3, 1.7]$. Noisy annotated bounding boxes are in red and the modified bounding boxes are in green.

still greatly reduced after correction.

These experimental results prove that our method can effectively reduce the influence of localization noise on the detector and the noise level of training annotations.

We further evaluate our proposed method on Gaussian-noise datasets, which are more similar to real-world noise. The results shown in Table 2 demonstrate that our method can also alleviate the impact of real-world noise on an object detector and outperform the baseline.

4.4 Additional Experiments on RefineDet

To demonstrate the generality of our proposal, we also applied our method on RefineDet, whose backbone is VGG16 pre-trained on Imagenet. We use a SGD optimizer to train the RefineDet on the artificial noisy datasets for 20,000 it-

Table 1 Test performance (mAP%) of Faster R-CNN trained on uniform-noise datasets.

Noise rate	Noise range			
		0.7-1.3	0.5-1.5	0.3-1.7
60%	Baseline	67.44	66.33	62.93
	Ours	67.84	66.61	64.96
80%	Baseline	67.32	63.86	59.46
	Ours	67.37	66.38	62.45
100%	Baseline	66.68	63.18	54.40
	Ours	67.60	64.52	58.56

Table 2 Test performance (mAP%) of Faster R-CNN trained on Gaussian-noise datasets.

Gaussian Distribution	Baseline	Ours
$\mu = 1, \sigma = 0.25$	65.09	66.08
$\mu = 1, \sigma = 0.30$	61.64	63.38
$\mu = 1, \sigma = 0.35$	59.59	62.72

erations with a weight decay of 5×10^{-4} , learning rate of 10^{-3} , momentum of 0.9 and batch size of 32. The learning rate decays to 10^{-4} and 10^{-5} at the 16,000-th and 18,000-th iteration, respectively. When training the model with our proposal, we start annotation refinement from the 14,000-th iteration. The hyperparameters w and t are set as 1 and 0.5 for all experiments, respectively.

Table 3 shows the results on uniform-noise datasets and

Table 3 Test performance (mAP%) of RefineDet trained on uniform-noise datasets.

Noise rate	Noise range				Clean
		0.7-1.3	0.5-1.5	0.3-1.7	
60%	Baseline	68.75	66.61	62.80	71.45
	Ours	68.78	67.52	64.76	
80%	Baseline	68.82	63.80	57.73	
	Ours	69.02	65.28	61.25	
100%	Baseline	68.57	61.17	51.25	
	Ours	68.83	62.81	57.21	

Table 4 Test performance (mAP%) of RefineDet trained on Gaussian-noise datasets.

Gaussian Distribution	Baseline	Ours
$\mu = 1, \sigma = 0.25$	65.27	66.86
$\mu = 1, \sigma = 0.30$	61.89	63.54
$\mu = 1, \sigma = 0.35$	58.14	61.52

Table 4 shows the results on Gaussian-noise datasets. Our method significantly alleviate the negative impact of noise, and the experimental results prove the generality of our proposal.

4.5 Quantitative Sensitivity Analysis

To examine the relationship between the hyperparameters and the performance of our proposal, we repeat the experiments of uniform noise datasets under different setting of the hyperparameters, as shown in Table 5 and Table 6. t and w denote the IoU threshold and weight in Eq. (1), respectively. According to the results, when the dataset is extremely noisy, setting a smaller threshold value can significantly obtain better results. For most other datasets, the effect of the IoU threshold on the accuracy is not obvious. The accuracy of our proposal is insensitive to the weight w .

5. Experiment on Real-World Dataset

In this section, we showed the effectiveness of our method on real-world dataset.

5.1 Dataset

We used Global Wheat Head Detection (GWHD) [14] dataset. This dataset contains 4,700 images and 190,000 annotated wheat heads collected from several countries around

Table 5 Test performance (mAP%) of Faster R-CNN trained on uniform-noise datasets using different hyperparameters.

Noise Range		0.7-1.3			0.5-1.5			0.3-1.7			clean
Noise Rate		60%	80%	100%	60%	80%	100%	60%	80%	100%	
$t = 0.3$	$w = 1$	67.73	67.33	67.50	66.39	66.10	64.68	64.83	63.41	61.46	70.32
	$w = 2$	67.49	67.42	66.99	66.65	65.92	64.68	65.17	63.28	61.31	
$t = 0.4$	$w = 1$	67.66	67.44	67.27	67.09	65.76	64.21	64.92	62.64	60.47	
	$w = 2$	67.51	67.46	67.08	66.84	66.16	64.23	64.86	63.09	61.13	
$t = 0.5$	$w = 1$	67.84	67.37	67.60	66.61	66.38	64.52	64.96	62.45	58.56	
	$w = 2$	67.68	67.38	67.33	67.02	65.51	65.10	64.95	62.80	59.75	
Best		67.84	67.46	67.60	67.09	66.38	65.10	65.17	63.41	61.46	
Baseline		67.44	67.32	66.68	66.33	63.86	63.18	62.93	59.46	54.40	

Table 6 Test performance (mAP%) of RefineDet trained on uniform-noise datasets using different hyperparameters.

Noise Range		0.7-1.3			0.5-1.5			0.3-1.7			clean
Noise Rate		60%	80%	100%	60%	80%	100%	60%	80%	100%	
$t = 0.3$	$w = 1$	68.86	68.95	68.81	67.39	64.90	63.43	65.51	62.05	58.36	71.45
	$w = 2$	68.10	68.69	68.97	67.08	64.85	62.89	65.75	62.06	58.30	
$t = 0.4$	$w = 1$	68.85	68.71	68.64	67.03	65.59	63.83	65.05	61.71	57.73	
	$w = 2$	68.46	68.84	68.31	67.27	64.19	63.16	65.26	61.56	58.18	
$t = 0.5$	$w = 1$	68.78	69.02	68.83	67.52	65.28	62.81	64.76	61.25	57.21	
	$w = 2$	68.53	68.41	68.55	66.81	65.11	63.36	64.63	61.09	56.97	
Best		68.86	69.02	68.97	67.52	65.59	63.83	65.75	62.06	58.36	
Baseline		68.75	68.82	68.57	66.61	63.80	61.17	62.80	57.73	51.25

Fig. 7 Visualization of the correction results of the GWHD dataset. The red bounding boxes are the annotations that are considered noisy, the white bounding boxes shows the annotations after modification.

the world at different growth stages with a wide range of genotypes. Part of the annotations in this dataset were machine-generated, and error-corrected by humans, then the annotations are a little noisy. We obtained the test dataset which were carefully made by humans. Thus, we utilized this dataset to evaluate our proposal.

5.2 Implementation

In this experiment, we fine-tune Faster R-CNN, whose backbone is ResNet101 [29] pre-trained on Imagenet, as the baseline. We use a SGD optimizer to train the network for 72 epochs with a weight decay of 5×10^{-4} and learning rate of 5×10^{-3} . The learning rate decays to 10^{-3} at the end of the 50-th epoch. When training the model with our proposal, we perform the refinement once because the noise level is small in this dataset. We use the prediction of the 50-th epoch to modify the original annotations by Algorithm 1, and the refined annotations were used to supervise the training of the remaining 22 epochs. The hyperparameters for modification, weight w and the threshold, are set as 9 and 0.8, respectively.

5.3 Results

Figure 7 shows the correction results of our method on the annotations of the GWHD dataset. The red bounding boxes indicate the original annotations in the GWHD dataset, and the white bounding boxes show the annotations after modification. The visualization shows that our method can refine the annotations with noticeable noise. Table 7 shows the test

Table 7 Test performance of Faster R-CNN trained on the GWHD dataset. AP and $mAP@.5:.95$ are the general metrics for object detection, and AP^* denotes the metric used in the kaggle competition based on GWHD dataset.

	AP^*_{best}	AP^*_{last}	AP^{50}	AP^{60}	AP^{70}	$mAP@.5:.95$
Baseline	70.22	68.78	87.79	79.84	62.69	47.91
Ours	71.17	70.71	87.95	80.70	64.74	48.92

performance on the test split of the GWHD dataset. AP^* follows the metric used in the kaggle competition[†] based on the GWHD dataset. Based on the refined annotations, the accuracy of the trained model has been improved.

6. Conclusion

In this study, we demonstrated the negative effects of noisy annotations on object detectors using artificial noisy datasets. We observed that a higher level of noise leads to a higher training loss. We further proposed an alternating refinement framework based on this finding to refine the noisy localization annotations during training. Our proposal is experimentally confirmed to be effective in reducing the effects of noise on the detector, especially when the noise is noticeable, by reducing the noise level in training noisy annotations.

Acknowledgements

This work was partially supported by JST CREST JP-MJCR1686 and JSPS KAKENHI 18H03254, Japan. The authors wish to thank Mr. Guo Wei for providing test split of the GWHD dataset.

References

- [1] J. Mao, Q. Yu, and K. Aizawa, “Noisy localization annotation refinement for object detection,” 2020 IEEE Int. Conf. Image Process. (ICIP), pp.2006–2010, 2020.
- [2] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol.39, no.6, pp.1137–1149, June 2016.
- [3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” CVPR, pp.580–587, 2014.
- [4] R. Girshick, “Fast r-cnn,” ICCV, pp.1440–1448, 2015.
- [5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” ICCV, pp.2961–2969, 2017.
- [6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” CVPR, pp.779–788, 2016.
- [7] J. Redmon and A. Farhadi, “Yolo9000: better, faster, stronger,” CVPR, pp.7263–7271, 2017.
- [8] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” arXiv preprint arXiv:1804.02767, 2018.
- [9] A. Bochkovskiy, C.Y. Wang, and H.Y.M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” arXiv preprint arXiv:2004.10934, 2020.
- [10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, and A.C. Berg, “Ssd: Single shot multibox detector,” ECCV, pp.21–37, 2016.

[†]<https://www.kaggle.com/c/global-wheat-detection>

- Springer, 2016.
- [11] T.Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C.L. Zitnick, "Microsoft coco: Common objects in context," *ECCV*, pp.740–755, Springer, 2014.
 - [12] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," *CVPR*, pp.248–255, 2009.
 - [13] D.P. Papadopoulos, J.R. Uijlings, F. Keller, and V. Ferrari, "Extreme clicking for efficient object annotation," *ICCV*, pp.4930–4939, 2017.
 - [14] E. David, S. Madec, P. Sadeghi-Tehran, H. Aasen, B. Zheng, S. Liu, N. Kirchgessner, G. Ishikawa, K. Nagasawa, M.A. Badhon, C. Pozniak, B. de Solan, A. Hund, S.C. Chapman, F. Baret, I. Stavness, and W. Guo, "Global wheat head detection (gwhd) dataset: a large and diverse dataset of high resolution rgb labelled images to develop and benchmark wheat head detection methods," *arXiv preprint arXiv:2005.02162*, 2020.
 - [15] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, "Learning from massive noisy labeled data for image classification," *CVPR*, pp.2691–2699, 2015.
 - [16] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang, "Bounding box regression with uncertainty for accurate object detection," *CVPR*, pp.2888–2897, 2019.
 - [17] P. Tang, X. Wang, X. Bai, and W. Liu, "Multiple instance detection network with online instance classifier refinement," *CVPR*, pp.2843–2851, 2017.
 - [18] I. Misra, A. Shrivastava, and M. Hebert, "Watch and learn: Semi-supervised learning of object detectors from videos," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp.3593–3602, 2015.
 - [19] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *IJCV*, vol.88, no.2, pp.303–338, 2010.
 - [20] S. Zhang, L. Wen, X. Bian, Z. Lei, and S.Z. Li, "Single-shot refinement neural network for object detection," *CVPR*, 2018.
 - [21] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M.S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, and S. Lacoste-Julien, "A closer look at memorization in deep networks," *ICML*, pp.233–242, 2017.
 - [22] I.J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.
 - [23] A. Vahdat, "Toward robustness against label noise in training deep discriminative neural networks," *NeurIPS*, pp.5596–5605, 2017.
 - [24] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," *CVPR*, pp.1944–1952, 2017.
 - [25] S. Sukhbaatar, J.B. Estrach, M. Paluri, L. Bourdev, and R. Fergus, "Training convolutional networks with noisy labels," *ICLR*, 2015.
 - [26] H. Zhang, M. Cisse, Y.N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.
 - [27] D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa, "Joint optimization framework for learning with noisy labels," *CVPR*, pp.5552–5560, 2018.
 - [28] S. Chadwick and P. Newman, "Training object detectors with noisy data," *2019 IEEE Intelligent Vehicles Symposium (IV)*, pp.1319–1325, 2019.
 - [29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp.770–778, 2016.

Jiafeng Mao received B.E. from the Huazhong University of Science and Technology in 2017. He is currently a M.S. student of Information Science and Technology at the University of Tokyo.

Qing Yu received B.S. in Information and Communication Engineering and M.S. in Interdisciplinary Studies in Information Science from the University of Tokyo in 2018 and 2020, respectively. He is currently a Ph.D. student of Information Science and Technology at the University of Tokyo.

Kiyoharu Aizawa received the B.E., the M.E., and the Dr. Eng. degrees in Electrical Engineering all from the University of Tokyo, in 1983, 1985, 1988, respectively. He is currently a Professor at Department of Information and Communication Engineering of the University of Tokyo. He was a Visiting Assistant Professor at University of Illinois from 1990 to 1992. His research interest is in image processing and multimedia applications. He received the 1987 Young Engineer Award and the 1990, 1998 Best

Paper Awards, the 1991 Achievement Award, 1999 Electronics Society Award from IEICE Japan, and the 1998 Fujio Frontier Award, the 2002 and 2009 Best Paper Award, and 2013 Achievement award from ITE Japan. He received the IBM Japan Science Prize in 2002. He is currently a Senior Associate Editor of *IEEE Trans. Image Processing*, and on Editorial Board of *ACM TOMM*, *APSIPA Transactions on Signal and Information Processing*, and *International Journal of Multimedia Information Retrieval*. He served as the Editor in Chief of *Journal of ITE Japan*, an Associate Editor of *IEEE Trans. Image Processing*, *IEEE Trans. CSVT* and *IEEE Trans. Multimedia*. He has served a number of international and domestic conferences; he was a General co-Chair of *ACM Multimedia 2012*. He is a council member of Science Council of Japan.