Learning with Noisy Class Labels for Instance Segmentation

Longrong Yang, Fanman Meng, Hongliang Li*, Qingbo Wu, and Qishang Cheng

School of Information and Communication Engineering,
University of Electronic Science and Technology of China
yanglr@std.uestc.edu.cn, {fmmeng, hlli, qbwu}@uestc.edu.cn,
cqs@std.uestc.edu.cn

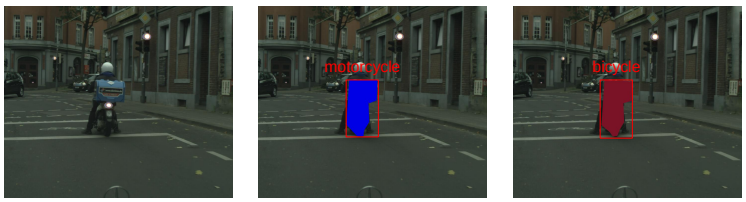
Abstract. Instance segmentation has achieved siginificant progress in the presence of correctly annotated datasets. Yet, object classes in large-scale datasets are sometimes ambiguous, which easily causes confusion. In addition, limited experience and knowledge of annotators can also lead to mislabeled object classes. To solve this issue, a novel method is proposed in this paper, which uses different losses describing different roles of noisy class labels to enhance the learning. Specifically, in instance segmentation, noisy class labels play different roles in the foreground-background sub-task and the foreground-instance sub-task. Hence, on the one hand, the noise-robust loss (e.g., symmetric loss) is used to prevent incorrect gradient guidance for the foreground-instance sub-task. On the other hand, standard cross entropy loss is used to fully exploit correct gradient guidance for the foreground-background sub-task. Extensive experiments conducted with three popular datasets (i.e., Pascal VOC, Cityscapes and COCO) have demonstrated the effectiveness of our method in a wide range of noisy class labels scenarios. Code will be available at: github.com/longrongyang/LNCIS.

Keywords: noisy class labels, instance segmentation, foreground-instance sub-task, foreground-background sub-task

1 Introduction

Datasets are of crucial to instance segmentation. Large-scale datasets with clean annotations are often required in instance segmentation. However, some classes show similar appearance and are easily mislabeled, as shown in Fig. 1. Meanwhile, some existing papers [11, 19, 24] also mention that inherent ambiguity of classes and limited experience of annotators can result in corrupted object class labels. These mislabeled samples inevitably affect the model training. Therefore, how to train accurate instance segmentation models in the presence of noisy class labels is worthy to explore.

* Corresponding author



(a) Image (b) Sample without Noise (c) Sample with Noise

Fig. 1. Example of noisy samples in the instance segmentation task. This example is selected from Cityscapes dataset [5]. In this example, object class *motocycle* is mislabeled as class *bicycle* by the annotator. We mainly discuss the noise on object class labels in this paper. Similar with methods in classification, datastes with lots of noise are produced by artificially corrupting labels.

In the classification task, label noise problem has been studied for a long time. Some existing methods apply the noise-robust loss (e.g., symmetric loss) to all samples to reduce gradients generated by noisy samples, such as [8, 29, 33]. These works have achieved promising results in the classification task. However, in the instance segmentation task, noisy class labels play different roles in the foreground-background sub-task (i.e., distinguishing foreground and background) and the foreground-instance sub-task (i.e., classifying different classes of foreground instances). From the perspective of the foreground-background sub-task, all class labels always provide correct guidance to the gradient update. Hence, if some key samples to the foreground-background sub-task are suppressed by the noise-robust loss in the gradient computation, the foreground-background sub-task is inevitably degenerated.

To solve this problem, we propose a novel method in this paper, which describes different roles of noisy class labels using diverse losses to enhance the learning. Firstly, some evidences provided in [1, 21, 32] show that models prone to fit clean and noisy samples in early and mature stages of training, respectively. Hence, in early stages of training, the classification loss remains unchanged. In mature stages of training, we observe that negative samples (i.e., samples belonging to background) are impossibly noisy and pesudo negative samples (i.e., positive samples misclassified as background) play key role in the foreground-background sub-task. Hence, cross entropy loss is applied to all negative samples and pesudo negative samples, to fully exploit correct gradient guidance provided by these samples for the foreground-background sub-task. Meanwhile, the noise-robust loss is applied to other samples for the foreground-instance sub-task. In addition, we also use loss values as the cue to detect and isolate some noisy samples, to further avoid the incorrect guidance provided by noisy class labels for the foreground-instance sub-task. This proposed method is verified on three well-known datasets, namely Pascal VOC [7], Cityscapes [5] and COCO [19]. Ex-

tensive experiments show that our method is effective in a wide range of noisy class labels scenarios.

2 Related Works

Learning with noisy class labels for the classification task: Different methods have been proposed to train accurate classification models in the presence of noisy class labels, which can be roughly divided into three categories. The first category is to improve the quality of raw labels by modeling the noises through directed graphical models [30], conditional random fields [27], neural networks [17,28] or knowledge graphs [18]. These methods usually require a set of clean samples to estimate the noise model, which limits their applicability. Hence, the joint optimization framework [26] does unsupervised sample relabeling with own estimate of labels. Meanwhile, Label Smoothing Regularization [23, 25] can alleviate over-fitting to noisy class labels by soft labels. The second category is to compensate for the incorrect guidance provided by noisy samples via modifying the loss functions. For example, Backward [22] and Forward [22] explicitly model the noise transition matrix to weight the loss of each sample. However, this method is hard to use because the noise transition matrix is not always availabel in practice [12]. Hence, some noise-robust loss functions are designed, such as MAE [8]. However, training a model with MAE converges slowly. To deal with this issue, Generalized Cross Entropy Loss [33] combines advantages of MAE and cross entropy loss by Box-Cox transformation [2]. Meanwhile, symmetric Cross Entropy Loss is proposed in [29], which applies the weighting sum of reverse cross entropy loss and cross entropy loss to achieve promising results in classification. These methods only require minimal intervention to existing algorithms and architectures. The third category is to introduce an auxiliary model. For example, a TeacherNet is used to provide a data-driven curriculum for a StudentNet by a learned sample weighting scheme in MentorNet [16]. To solve the accumulated error issue in MentorNet, Co-teaching [13] maintains two models simultaneously during training, with one model learning from the another model's most confident samples. Furthermore, Co-teaching+ [31] keep two networks diverged during training to prevent Co-teaching reducing to the self-training MentorNet in function.

These methods suppose that noisy class labels inevitably degenerate the model accuracy, which is suitable for the classification task, but is invalid for the instance segmentation task with multiple sub-tasks such as foreground-background and foreground-instance. It is the fact that noisy labels play different roles in the two sub-tasks, which need be treated differently.

Instance segmentation: Some instance segmentation models have been proposed in the past few years [3,6,14,15,20]. Based on the segmentation manner, these methods can be roughly divided into two categories. The first one is driven by the success of the semantic segmentation task, which firstly predicts the class of each pixel, and then groups different instances, such as GMIS [20] and DLF [6]. The second one connects strongly with the object detection task, such

as Mask R-CNN [14], Mask Scoring R-CNN [15] and HTC [3], which detects object instances firstly, and then generates masks from the bounding boxes. Among these methods, Mask R-CNN [14] selected as the reference backbone for the task of instance-level segmentation in this paper consists of four steps. The first one is to extract features of images by CNNs. The second one is to generate proposals by RPN [10]. The third one is to obtain the classification confidence and the bounding box regression. Finally, segmentation masks are generated inside of bounding boxes by the segmentation branch.

Although Mask R-CNN [14] has achieved promising results for the instance segmentation task, it is based on clean annotations. When there are noisy labels, its performance drops significantly. In contrast to the classification task, Mask R-CNN [14] has multiple sub-tasks with different roles and classification losses. From the perspective of the foreground-background sub-task, noisy class labels still provide correct guidance. Meanwhile, in instance segmentation, proposal generation and mask generation are only related with the foreground-background sub-task. Hence, the binary classification losses in RPN [10] and the segmentation branch remains unchanged. In this paper, we focus on the multi-class classification loss in the box head, which is related with the foreground-background sub-task and the foreground-instance sub-task, simultaneously.

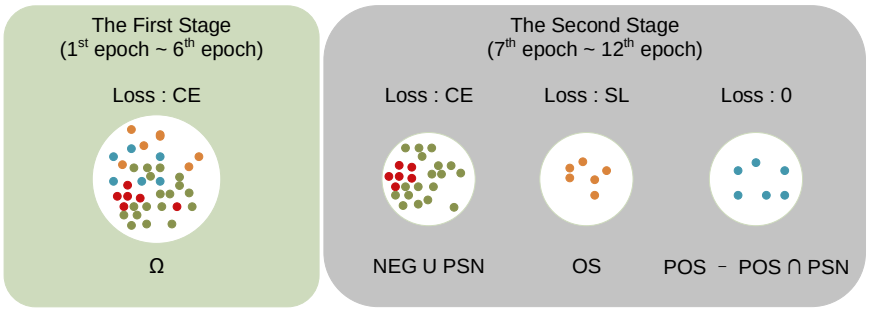


Fig. 2. Losses of different types of samples. Ω denotes the total sample space. $NEG \cup PSN$ denotes all negative samples and pseudo negative samples. $POS - POS \cap PSN$ denotes potential noisy samples classified foreground. OS denotes other samples. CE and SL denote standard cross entropy loss and symmetric loss, respectively.

3 Methodology

The multi-class classification in instance segmentation consists of the foreground-background sub-task and the foreground-instance sub-task. In general, it can be formulated as the problem to learn a classifier $f_\theta(x)$ from a set of training samples $D = (x_i, y_i)_{i=1}^N$ with $y_i \in \{0, 1, ..., K\}$. In instance segmentation, the sample x_i

corresponds to an image region rather than an image. y_i is the class label of the sample x_i and can be noisy. For convenience, we assign 0 as the class label of samples belonging to background. Meanwhile, suppose the correct class label of the sample x_i is $y_{c,i}$. In this paper, we focus on the noise on object class labels and 0 is not a true object class label in datasets, so $p(y_i = 0|y_{c,i} \neq 0) = p(y_i \neq 0|y_{c,i} = 0) = 0$. By a loss function, e.g., multi-class cross entropy loss, the foreground-background sub-task and the foreground-instance sub-task are optimized simultaneously:

$$l_{ce} = \frac{1}{N}\sum_{i=1}^{N} l_{ce,i} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=0}^{K} q(k|x_i) log p(k|x_i) \tag{1}$$

where $p(k|x_i)$ denotes classification confidence of each class $k \in \{0, 1, ..., K\}$ for the sample x_i, and $q(k|x_i)$ denotes the one-hot encoded label, so $q(y_i|x_i) = 1$ and $q(k|x_i) = 0$ for all $k \neq y_i$.

Cross entropy loss is sensitive to label noise. Some existing methods in classification use the noise-robust loss (e.g., symmetric loss) to replace cross entropy loss to deal with this problem. However, the noise-robust loss leads to reduced gradients of some samples, which degenerates the foreground-background sub-task in instance segmentation. To solve this problem, we describes different roles of noisy class labels using diverse losses, as shown in Fig. 2.

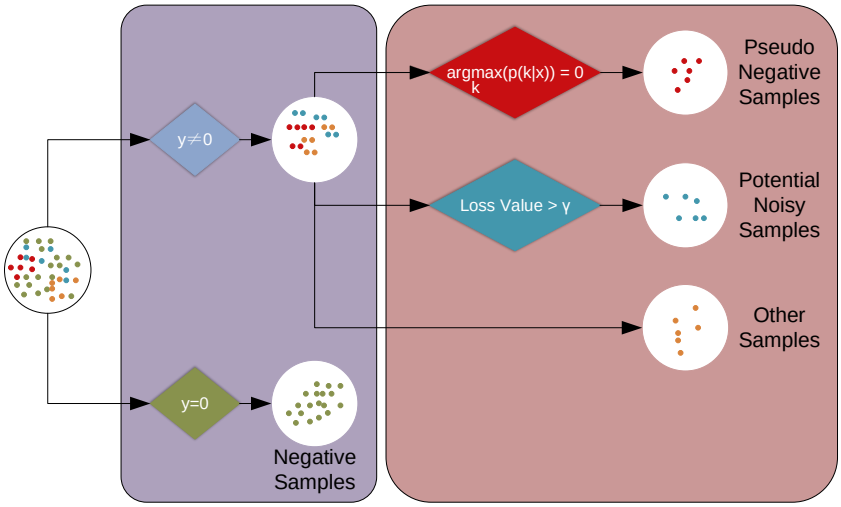


Fig. 3. Division of samples in this paper. Here, y is the class label of the sample x. γ is a hyper-parameter that can adjust. $p(k|x)$ is the confidence of x on class k. Some samples possibly belong to pseudo negative samples and potential noisy samples, simultaneously.

3.1 Division of Samples

Firstly, as shown in Fig. 3, all samples are roughly divided into two types: positive samples and negative samples. In Mask R-CNN [14], positive samples refer to the samples whose IoUs (Intersection over Union) with the corresponding ground truths are above 0.5. Hence, it is generally considered that positive samples belong to foreground (i.e., $y \neq 0$) and negative samples belong to background (i.e., $y = 0$).

Furthermore, positive samples are divided into three types: pseudo negative samples (PSN), potential noisy samples (PON) and other samples (OS). Here, we define positive samples which are misclassified as background as pseudo negative samples. Hence, it is easy to know that, for a pseudo negative samples x, $argmax_k(p(k|x)) = 0$. In addition, as shown in Fig. 3, we define samples whose loss values $l_{ce} > \gamma$ as potential noisy samples. According to our stastics in Fig. 5, in instance segmentation, noisy samples usually have larger loss values than clean samples in mature stages of training. From Fig. 5, it can be seen that 88.5% of noisy samples have loss values $l_{ce} > 6.0$ while only 2.31% of clean samples have loss values $l_{ce} > 6.0$. Subjective examples are also given in Fig. 4 to explain the difference of different samples.

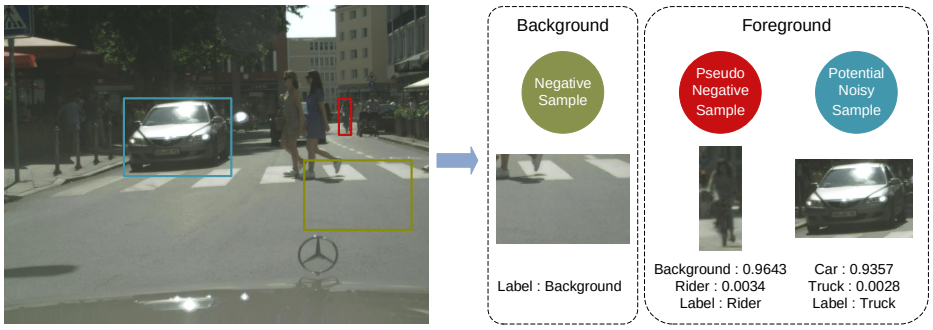


Fig. 4. Subjective examples to explain the difference of different samples. Car : 0.9357 denotes that the confidence of this sample is 0.9357 on class Car.

3.2 Classification Loss

In early stages of training, models favor learning clean samples, while hindering learning noisy samples [32]. As models converge to higher accuracy, noisy samples gradually contribute more to the gradient update. In mature stages of training, models prone to fit in noisy samples. Hence, in early stages of training (the first stage), the classification loss remains unchanged (i.e., cross entropy loss is applied to all samples). Suppose total sample numbers of a batch are N. Classification loss of this batch in the first stage can be described as:

$$Loss_1 = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=0}^{K} q(k|x_i) log p(k|x_i) \tag{2}$$

where $Loss_1$ denotes the multi-class classification loss in the first stage. Suppose E_1 and E are epoch numbers of the first stage and total epoch numbers, respectively. Under different noise rates, we empirically derive the relation between E and E_1:

$$E_1 = \frac{1}{2}E, \quad s.t.\ \forall \eta \tag{3}$$

where η denotes noise rate. In general, $E = 12$ and $E_1 = 6$.

In mature stages of training (the second stage), the noise-robust loss needs be introduced. However, this loss leads to reduced gradients of some key samples to the foreground-background sub-task in instance segmentation. Hence, cross entropy loss still needs to be used, to prevent the noise-robust loss from degenerating the foreground-background sub-task. Different losses are applied to different types of samples to yield a good compromise between fast convergence and noise robustness.

Firstly, in instance segmentation, noisy class labels only exist in partial foreground regions. Hence, we think that potential noisy labels do not exist in any negative samples. To fully exploit correct gradient information provided by these samples for the foreground-background sub-task, standard cross entropy loss is applied to all negative samples. Secondly, it is clear that the noise on object class labels does not change this fact that a positive sample belongs to foreground. Therefore, if a positive sample is misclassified as background, this sample still plays key role in the foreground-background sub-task even if it is noisy. For this reason, standard cross entropy loss is also applied to all pseudo negative samples. In addition, potential noisy samples classified as foreground can be isolated in the gradient computation, to further aviod incorrect guidance provided by noisy class labels to the foreground-instance sub-task.

Suppose total sample numbers of a batch are N. Meanwhile, there are N_1 negative samples, N_2 pseudo negative samples, N_3 potential noisy samples classified foreground and N_4 other samples in this batch. $N = N_1 + N_2 + N_3 + N_4$. In mature stages of training, classification loss of this batch can be described as:

$$Loss_2 = -\frac{1}{N}[\sum_{i=1}^{N_1+N_2}\sum_{k=0}^{K} q(k|x_i) log p(k|x_i) + \sum_{m=1}^{N_3} 0 + \sum_{j=1}^{N_4}(-l_{sl,j})] \tag{4}$$

where $Loss_2$ denotes the multi-class classification loss in the second stage. $l_{sl,j}$ denotes a symmetric loss function which is robust to noise.

3.3 Reverse Cross Entropy Loss

In this paper, we select reverse cross entropy loss proposed in [29] as the symmetric loss $l_{sl,i}$. The reverse cross entropy loss is defined as:

$$l_{rce,i} = -\sum_{k=0}^{K} p(k|x_i) log q(k|x_i) \tag{5}$$

where $l_{rce,i}$ denotes the reverse cross entropy loss for the sample x_i. This kind of loss is robust to noise but takes longer time to converge [29]. Meanwhile, there is also a compromise in accuracy due to the increased difficulty to learn useful features. In this paper, the clipped replacement A of $log0$ is set to -4.

4 Theoretical Analyses

4.1 Noise Robustness

We explain the noise robustness of reverse cross entropy loss from the perspective of symmetric loss. For any classifier f, $R(f)$ or $R^{\eta}(f)$ denotes the risk of f under clean class labels or noisy class labels (noise rate η), respectively. Suppose f^* and f^*_{η} are the global minimizers of $R(f)$ and $R^{\eta}(f)$, respectively. Let $\mathcal{X} \subset R^d$ be the feature space. A symmetric loss function l is defined as:

$$\sum_{y=0}^{K} l(f(x), y) = C, \ \forall x \in \mathcal{X}, \ \forall f. \tag{6}$$

where C is a constant. In [9], it has been proven that if loss function is symmetric and noise rate $\eta < \frac{K}{K+1}$, then under symmetric label noise, for $\forall\ f$, $R^{\eta}(f^*) - R^{\eta}(f) = (1 - \frac{\eta(K+1)}{K})(R(f^*) - R(f)) \leqslant 0$. Hence, $f^*_{\eta} = f^*$ and l is noise robust. Moreover, it can also be proven, if $R(f^*) = 0$, that l is noise robust under asymmetric noise. Meanwhile, according to Eq. 5, we can derive:

$$\begin{aligned}
\sum_{y=0}^{K} l_{rce} &= -\sum_{y=0}^{K}\sum_{k=0}^{K} p(k|x) log q(k|x) \\
&= -\sum_{y=0}^{K}(\sum_{k\neq y}^{K} p(k|x) log 0 + p(y|x) log 1) \\
&= -\sum_{y=0}^{K}[1 - p(y|x)] log 0 = -KA
\end{aligned}$$

where $-KA$ is a constant when class numbers K and A ($log0 = A$) are given. Hence, reverse cross entropy loss is symmetric and noise robust.

4.2 Gradients

Gradients of reverse cross entropy loss and cross entropy loss have been derived in [29]. Based on this, we can explain why models favor learning clean samples in early stages of training, while hindering learning noisy samples. This is not discussed in [29]. For brevity, we denote p_k, q_k as abbreviations for $p(k|x)$ and $q(k|x)$. We focus on a sample $\{x, y\} \subset D$. The gradient of reverse cross entropy loss with respect to the logit z_j can be derived as:

$$\frac{\partial l_{rce}}{\partial z_j} = \begin{cases} Ap_j - Ap_j^2, & q_j = q_y = 1 \\ -Ap_j p_y, & q_j = 0 \end{cases} \tag{7}$$

The gradient of cross entropy loss l_{ce} is:

$$\frac{\partial l_{ce}}{\partial z_j} = \begin{cases} p_j - 1, & q_j = q_y = 1 \\ p_j, & q_j = 0 \end{cases} \tag{8}$$

Analysis. As shown in Eq. 8, for cross entropy loss, if $q_j = 1$, samples with smaller p_j contribute more to the graident update. Based on this, it is clear that models gradually prone to fit noisy samples as models converge to higher accuracy. To clarify this, suppose sample A and sample B belong to the same class C_1. p_{a,c_1} denotes that the classification confidence of sample A on class C_1. Meanwhile, suppose class labels of sample A and B are C_1 and C_2, respectively ($C_1 \neq C_2$). At the beginning of the training, $p_{a,c_1} \approx p_{b,c_2}$ hence their contribution to the gradient computation is approximately equal. When the training continues to later stages of training, because the accuracy of models increases, p_{a,c_1} increases and p_{b,c_2} decreases. As a result, gradients generated by noisy samples become larger and gradients generated by clean samples become smaller. When noisy samples contribute more to the gradient computation than clean samples, model begins to prone to fit noisy samples.

Secondly, the weakness of reverse cross entropy loss can also be explained from the respect of gradients. As shown in Eq. 7, if $q_j = 1$, gradients of reverse cross entropy loss are symmetric about $p_j = 0.5$. This means that, if p_j of a sample is close to 0, the gradient generated by this sample is also close to 0. This is the reason why this loss is robust to noise. However, this also leads to reduced gradients of clean samples whose p_j is close to 0 and these samples usually play key role in training.

5 Experiments

5.1 Datasets and Noise Settings

Pascal VOC dataset: On Pascal VOC 2012 [7], the *train* subset with 5718 images and the *val* subset with 5823 images are used to train the model and evaluate the performance, respectively. There are 20 semantic classes on Pascal

VOC dataset [7]. All objective results are reported following the COCO-style metrics which calculates the average AP across IoU thresholds from 0.5 to 0.95 with an interval of 0.05.

COCO dataset: COCO dataset [19] is one of the most challenging datasets for the instance segmentation task due to the data complexity. It consists of 118,287 images for training (*train-2017*) and 20,288 images for test (*test-dev*). There are 80 semantic classes on COCO dataset [19]. We train our models on *train-2017* subset and report objective results on *test-dev* subset. COCO standard metrics are used in this paper, which keeps the same with traditional instance segmentation methods.

Cityscapes dataset: On Cityscapes dataset [5], the fine training set which has 2975 images with fine annotations is used to train the model. The validation set has 500 images, which is used to evaluate the performance of our method. Eight semantic classes are annotated with instance masks.

Noise settings: Noisy datasets are produced by artificially corrupting object class labels. Similar with noise settings in classification, there are mainly two types of noise in this paper: symmetric (uniform) noise and asymmetric (class-conditional) noise. If the noise is conditionally independent of correct class labels, the noise is named as symmetric or uniform noise. Class labels with symmetric noise are generated by flipping correct class labels of training samples to a different class with uniform random probability η. For class labels with asymmetric noise, similar with [22,33], flipping labels only occurs within specific classes which are easily misclassified, for example, for VOC dataset, flipping $Bird \rightarrow Aeroplane$, $Diningtable \rightarrow Chair$, $Bus \rightarrow Car$, $Sheep \rightarrow Horse$, $Bicyce \leftrightarrow Motobike$, $Cat \leftrightarrow Dog$; for Cityscapes dataset, flipping $Person \leftrightarrow Rider$, $Bus \leftrightarrow Truck$, $Motorcycle \leftrightarrow Bicycle$; for COCO dataset, 80 classes are grouped into 12 super-classes based on the COCO standard, then flipping between two randomly selected sub-classes within each super-class. Super-class $Person$ is not flipped because this super-class only has a sub-class.

5.2 Implementation Details

Hyper-parameters: For COCO dataset, the overall batch size is set to 16. The initial learning rate is set to 0.02. We train for 12 epoches ($1\times$) and the learning rate reduces by a factor of 10 at 8 and 11 epoches. For Cityscapes dataset, the overall batch size is set to 2. The initial learning rate is set to 0.0025. We train for 64 epoches and the learning rate reduces by a factor of 10 at 48 epoches. For Pascal VOC dataset, the overall batch size is set to 4. The initial learning rate is set to 0.005. We train for 12 epoches and the learning rate reduces by a factor of 10 at 9 epoches. Other hyper-parameters follow the settings in MMdetection [4].

In our method, we set $\gamma = 6.0$ for all datasets. For symmertric noise, we test varying noise rates $\eta \in \{20\%, 40\%, 60\%, 80\%\}$, while for asymmetric noise, we test varying noise rates $\eta \in \{20\%, 40\%\}$.

Table 1. The results on Pascal VOC dataset. 0% denotes no artificially corrupting class labels. We report the mAPs of all methods. The best results are in bold.

Methods		Symmetric Noise				Asymmetric Noise	
Noise Rates(η)	0%	20%	40%	60%	80%	20%	40%
LSR [23]	36.7	34.5	31.9	27.0	20.3	35.8	33.2
JOL [26]	32.7	22.7	11.5	3.8	3.8	24.6	24.3
GCE [33]	38.7	35.3	32.9	26.1	15.5	36.0	32.3
SCE [29]	39.4	34.6	32.1	27.9	21.2	37.5	34.9
CE [14]	39.3	34.2	31.5	27.1	20.7	36.8	34.5
Our Method	**39.7**	**38.5**	**38.1**	**33.8**	**25.5**	**37.8**	**35.1**

Table 2. The results on COCO *test-dev* subset.

Methods		Symmetric Noise				Asymmetric Noise	
Noise Rates(η)	0%	20%	40%	60%	80%	20%	40%
SCE [29]	32.3	31.5	29.9	28.2	22.0	32.1	31.6
CE [14]	**34.2**	31.3	29.3	27.1	21.7	31.9	31.3
Our Method	33.7	**33.1**	**31.3**	**30.8**	**26.6**	**33.3**	**33.0**

5.3 Main Results

Baselines: How to effectively train an instance segmentation model in the presence of noisy class labels is never discussed in existing papers. Hence, we mainly compare our method with some methods [23, 26, 29, 33] proposed in the classification task as well as the standard CE loss [14]: (1) LSR [23]: training with standard cross entropy loss on soft labels; (2) JOL [26]: training with the joint optimization framework. Here, α is set to 0.8 and β is set to 1.2; (3) GCE [33]: training with generalized cross entropy loss. Here, q is set to 0.3; (4) SCE [29]: training with symmetric cross entropy loss $l_{sce} = \alpha l_{ce} + \beta l_{rce}$. Here, α is set to 1.0 and β is set to 0.1; (5) CE [14]: training with standard cross entropy loss. Based on our experiments, we select the best hyper-parameter settings for methods proposed in classification. Note we only change the multi-class classification loss in box head. Mask R-CNN [14] is used as the instance segmentation model and the backbone is ResNet-50-FPN in all methods.

Pascal VOC dataset: The objective results are reported in Table 1. Compared with [14], methods proposed in classification bring marginal accuracy increase (below 2% under different noise rates) if directly applied to instance segmentation. However, our method can generate a substantial increase in performance by using different losses describing different roles of noisy class labels. Specifically, compared with [14], the accuracy increases 4.3%, 6.6%, 6.7% and 4.8% for 20%, 40%, 60% and 80% of symmetric label noise, respectively. Under 20% and 40% asymmetric noise, the accuracy increases by 1.0% and 0.6%, respectively. It can be seen that our method yields an ideal accuracy under different noise rates.

COCO dataset: The objective results are reported in Table 2. Compared with [14], the accuracy increases 1.8%, 3.1%, 4.2% and 4.9% for 20%, 40%, 60%

Table 3. The results on Cityscapes dataset.

Methods		Symmetric Noise				Asymmetric Noise	
Noise Rates(η)	0%	20%	40%	60%	80%	20%	40%
SCE [29]	30.2	26.1	22.3	12.6	9.3	28.0	18.6
CE [14]	32.5	26.0	21.0	13.3	11.6	29.8	18.9
Our Method	**32.7**	**30.8**	**29.1**	**19.1**	**15.2**	**30.9**	**21.3**

and 80% of symmetric label noise, respectively. Under 20% and 40% asymmetric noise, the accuracy increases by 1.4% and 1.7%, respectively.

Cityscapes dataset: The objective results are reported in Table 3. Compared with [14], the accuracy increases 4.8%, 8.1%, 5.8% and 3.6% for 20%, 40%, 60% and 80% of symmetric label noise, respectively. Under 20% and 40% asymmetric noise, the accuracy increases by 1.1% and 2.4%, respectively.

5.4 Discussion

Component ablation study: Our component ablation study from the baseline gradually to all components incorporated is conducted on Pascal VOC dataset [7] and noise rate $\eta = 40\%$. We mainly discuss:

(*i*) **CE**: The baseline. The model is trained with standard cross entropy loss;

(*ii*) **ST**: Stage-wise training. We apply cross entropy loss and reverse cross entropy loss to all samples in early and mature stages of training, respectively.

(*iii*) **N** & **PSN**: The key contribution in this paper. Cross entropy loss is applied to all negative samples and pseudo negative samples. Meanwile, reverse cross entropy loss is applied to other samples;

(*iv*) **N** & **PSN** & **PON**: Cross entropy loss is applied to all negative samples and pseudo negative samples. Meanwhile, potential noisy samples classified as foreground are isolated in the gradient computation.

(*v*) **ST** & **N**: Stage-wise training is applied. Meanwhile, cross entropy loss is applied to negative samples in mature stage of training;

(*vi*) **ST** & **N** & **PSN**: Stage-wise training is applied. Meanwhile, cross entropy loss is applied to all negative samples and pseudo negative samples in mature stage of training;

(*vii*) **ST** & **N** & **PSN** & **PON**: Stage-wise training is applied. Meanwhile, cross entropy loss is applied to all negative samples and pseudo negative samples in mature stage of training. In addition, in mature stage of training, potential noisy samples classified as foreground are isolated in the gradient computation.

The results of ablation study are reported in Table 4. Firstly, the key strategy in this paper (i.e., using different losses to describe different roles of noisy class labels) brings 5.8% higher accuracy than the baseline, which shows that this strategy is greatly important in instance segmentation. Secondly, stage-wise training can bring 2.8% higher accuracy than the baseline. However, if already considering special properties of negative samples and pseudo negative samples, stage-wise training can only bring about 0.6% higher accuracy. This means, the

Table 4. Component ablation study. *CE* denotes the baseline. *ST* denotes stage-wise training. *N* denotes applying cross entropy loss to all negative samples. *PSN* denotes applying cross entropy loss to all pseudo negative samples. *PON* denotes isolating all potential noisy samples classified as foreground.

CE	ST	N	PSN	PON	AP	AP_{50}	AP_{75}
√	-	-	-	-	31.5	57.4	31.0
	√				34.3	59.9	35.1
		√	√		37.3	63.5	39.0
		√	√	√	36.5	63.2	37.4
	√	√			37.4	64.7	39.0
	√	√	√		37.9	**64.7**	38.0
	√	√	√	√	**38.1**	64.5	**40.0**

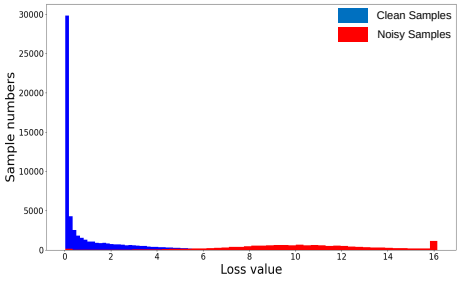
main reason that stage-wise training works in instance segmentation is factually to fully exploit correct gradient information for the foreground-background subtask. Thirdly, using loss values as the cue to identify noisy samples (i.e., PON) brings marginal accuracy increase (about 0.2%), and stage-wise training (i.e., ST) should be applied simultaneously when PON is applied.

The relation between E and E_1: Stage-wise training applies cross entropy loss to all samples in early stages of training, to speed the convergence of models. In mature stages of training, different losses are applied to different samples to yield a good compromise between fast convergence and noise robustness. We conduct some experiments about different E and E_1 under noise rate $\eta = 40\%$ in Table 5. From Table 5, it can be seen that $E_1 = 0.5E$ is the best setting.

Table 5. Study about the relation between E and E_1. E_1 and E are epoch numbers of the first stage and total epoch numbers, respectively.

	AP	AP_{50}	AP_{75}
$E = 12, E_1 = 3$	37.2	63.7	39.0
$E = 12, E_1 = 6$	**38.1**	**64.5**	**40.0**
$E = 12, E_1 = 9$	34.6	61.1	34.8
$E = 18, E_1 = 6$	36.2	62.2	37.9
$E = 18, E_1 = 9$	**38.5**	**66.0**	**39.8**
$E = 18, E_1 = 12$	38.2	66.0	39.7
$E = 18, E_1 = 15$	32.5	58.8	33.3

Hyper-parameter analysis: In our method, γ is a hyper-parameter that need be discussed. Positive samples whose loss values $l_{ce} > \gamma$ are named as potential noisy samples. We observe that for a sample, if $p_j = 0.01$ and $q_j = 1$, cross entropy loss $l_{ce} = 4.5052$. Meanwhile, if $p_j = 0.9$, $l_{ce} = 0.1054$. Hence, we think that γ must satisfy $\gamma \geq 5.0$ to identify noisy samples, which also fits statistics shown in Fig. 5. Several experiments are conducted about γ in Table 6. In our setting, we set $\gamma = 6.0$ for all datasets and all noise rates.



Sample Types	Numbers		Numbers	Percentage
Clean Samples	39436	Loss Value > 6.0	909	**2.31%**
		Loss Value < 6.0	38527	97.69%
Noisy Samples	9173	Loss Value > 6.0	8118	**88.5%**
		Loss Value < 6.0	1055	11.5%

Fig. 5. Loss values l_{ce} and statistics of samples after the sixth epoch. We sample about 50000 samples.

Table 6. The settings of γ. γ controls how many samples are identified as potential noisy samples.

	AP	AP_{50}	AP_{75}
$\gamma = 4.0$	37.5	64.0	37.8
$\gamma = 5.0$	38.0	**64.9**	39.7
$\gamma = 6.0$	**38.1**	64.5	**40.0**
$\gamma = 7.0$	38.0	64.7	40.0
$\gamma = 8.0$	37.2	62.6	38.1

6 Conclusion

In this paper, we propose a novel method to effectively train an instance segmentation model whose performance is robust under noisy supervision. Our key strategy is to use different losses describing different roles of noisy class labels. Based on this, correct gradient information is fully exploited for the foreground-background sub-task and incorrect guidance provided by noisy samples is avoided for the foreground-instance sub-task. We have conducted sufficient experiments on three well-known datasets (i.e., Pascal VOC, Cityscapes and COCO). The results show the superiority of our method in various noisy class labels scenarios.

Acknowledgement. This work was supported in part by National Natural Science Foundation of China (No.61831005, 61525102, 61871087 and 61971095).

References

1. Arpit, D., Jastrzbski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M.S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al.: A closer look at memorization in deep networks. In: ICML (2017)
2. Box, G.E.P., Cox, D.R.: An analysis of transformations. Journal of the Royal Statistical Society: Series B (Methodological) **26**(2), 211–243 (1964)
3. Chen, K., Pang, J., Wang, J., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Shi, J., Ouyang, W., Loy, C.C., Lin, D.: Hybrid task cascade for instance segmentation. In: CVPR (2019)
4. Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., et al.: Mmdetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155 (2019)
5. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3213–3223 (2016)
6. De Brabandere, B., Neven, D., Van Gool, L.: Semantic instance segmentation with a discriminative loss function. arXiv preprint arXiv:1708.02551 (2017)
7. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision **88**(2), 303–338 (2010)
8. Ghosh, A., Kumar, H., Sastry, P.: Robust loss functions under label noise for deep neural networks. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
9. Ghosh, A., Manwani, N., Sastry, P.: Making risk minimization tolerant to label noise. Neurocomputing **160**, 93–107 (2015)
10. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1440–1448 (2015)
11. Gygli, M., Ferrari, V.: Fast object class labelling via speech. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5365–5373 (2019)
12. Han, B., Yao, J., Niu, G., Zhou, M., Tsang, I., Zhang, Y., Sugiyama, M.: Masking: A new perspective of noisy supervision. In: Advances in Neural Information Processing Systems. pp. 5836–5846 (2018)
13. Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., Sugiyama, M.: Co-teaching: Robust training of deep neural networks with extremely noisy labels. In: Advances in Neural Information Processing Systems. pp. 8527–8537 (2018)
14. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Computer Vision (ICCV), 2017 IEEE International Conference on. pp. 2980–2988. IEEE (2017)
15. Huang, Z., Huang, L., Gong, Y., Huang, C., Wang, X.: Mask scoring r-cnn. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6409–6418 (2019)
16. Jiang, L., Zhou, Z., Leung, T., Li, L.J., Fei-Fei, L.: Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. arXiv preprint arXiv:1712.05055 (2017)
17. Lee, K.H., He, X., Zhang, L., Yang, L.: Cleannet: Transfer learning for scalable image classifier training with label noise. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5447–5456 (2018)
18. Li, Y., Yang, J., Song, Y., Cao, L., Luo, J., Li, L.J.: Learning from noisy labels with distillation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1910–1918 (2017)

19. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision. pp. 740–755. Springer (2014)
20. Liu, Y., Yang, S., Li, B., Zhou, W., Xu, J., Li, H., Lu, Y.: Affinity derivation and graph merge for instance segmentation. In: Proceedings of the European Conference on Computer Vision. pp. 686–703. Springer (2018)
21. Ma, X., Wang, Y., Houle, M.E., Zhou, S., Erfani, S.M., Xia, S.T., Wijewickrema, S., Bailey, J.: Dimensionality-driven learning with noisy labels. arXiv preprint arXiv:1806.02612 (2018)
22. Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., Qu, L.: Making deep neural networks robust to label noise: A loss correction approach. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1944–1952 (2017)
23. Pereyra, G., Tucker, G., Chorowski, J., Kaiser, Ł., Hinton, G.: Regularizing neural networks by penalizing confident output distributions. arXiv preprint arXiv:1701.06548 (2017)
24. Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., Li, J., Sun, J.: Objects365: A large-scale, high-quality dataset for object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 8430–8439 (2019)
25. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2818–2826 (2016)
26. Tanaka, D., Ikami, D., Yamasaki, T., Aizawa, K.: Joint optimization framework for learning with noisy labels. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5552–5560 (2018)
27. Vahdat, A.: Toward robustness against label noise in training deep discriminative neural networks. In: Advances in Neural Information Processing Systems. pp. 5596–5605 (2017)
28. Veit, A., Alldrin, N., Chechik, G., Krasin, I., Gupta, A., Belongie, S.: Learning from noisy large-scale datasets with minimal supervision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 839–847 (2017)
29. Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., Bailey, J.: Symmetric cross entropy for robust learning with noisy labels. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 322–330 (2019)
30. Xiao, T., Xia, T., Yang, Y., Huang, C., Wang, X.: Learning from massive noisy labeled data for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2691–2699 (2015)
31. Yu, X., Han, B., Yao, J., Niu, G., Tsang, I., Sugiyama, M.: How does disagreement help generalization against label corruption? In: International Conference on Machine Learning. pp. 7164–7173 (2019)
32. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning requires rethinking generalization. In: ICLR (2017)
33. Zhang, Z., Sabuncu, M.: Generalized cross entropy loss for training deep neural networks with noisy labels. In: Advances in Neural Information Processing Systems. pp. 8778–8788 (2018)