Towards A Category-extended Object Detector without Relabeling or Conflicts

Bowen Zhao1,† Chen Chen2,† Wanpeng Xiao2 Xi Xiao1,3 Qi Ju2,3 Shutao Xia1,3,✉
1Tsinghua University 2Tencent TEG AI 3Peng Cheng Laboratory
zbwl8@mails.tsinghua.edu.cn, {xiaox, xiast}@sz.tsinghua.edu.cn
{beckhamchen, wanpengxiao, damonju}@tencent.com

Abstract

Object detectors are typically learned based on fully-annotated training data with fixed pre-defined categories. However, not all possible categories of interest can be known beforehand, as classes are often required to be increased progressively in many realistic applications. In such scenario, only the original training set annotated with the old classes and some new training data labeled with the new classes are available. In this paper, we aim at leaning a strong unified detector that can handle all categories based on the limited datasets without extra manual labor. Vanilla joint training without considering label ambiguity leads to heavy biases and poor performance due to the incomplete annotations. To avoid such situation, we propose a practical framework which focuses on three aspects: better base model, better unlabeled ground-truth mining strategy and better retraining method with pseudo annotations. First, a conflict-free loss is proposed to obtain a usable base detector. Second, we employ Monte Carlo Dropout to calculate the localization confidence, combined with the classification confidence, to mine more accurate bounding boxes. Third, we explore several strategies for making better use of pseudo annotations during retraining to achieve more powerful detectors. Extensive experiments conducted on multiple datasets demonstrate the effectiveness of our framework for category-extended object detectors.

1. Introduction

Object detection is a fundamental and essential computer vision problem in various application areas, such as self-driving [12], medicine [36] and security [1], etc. Owing to the development of deep neural networks, more and more powerful detectors are proposed [10, 32, 24, 22, 31, 35]. A standard object detector is typically trained on a pre-prepared fully annotated training dataset with fixed categories. However, in many real-world applications, we cannot know all possible categories of interest beforehand, so it is often desired to add new object classes progressively. For instance, a detector in use was trained on an original dataset (e.g., COCO [23], which consists of 80 categories). Now, it is required to detect a new class “human face” simultaneously while only some new training data labeled with “human face” is additionally provided (like WIDER FACE [39]). As shown in Figure 1, many faces in the original training data are not labeled, similarly, the instances belonging to the original categories (e.g., “person”) are not annotated in the new dataset.

Based on the limited datasets, how to train a unified detector that can manage all categories? There are two straightforward ways to achieve the above goal, as shown in Figure 1. One is relabeling, which transforms the problem into a standard object detection task. However, it’s very expensive to relabel all missing classes in the original and new datasets. For instance, a detector in use was trained on an original dataset (e.g., COCO [23], which consists of 80 categories). Now, it is required to detect a new class “human face” simultaneously while only some new training data labeled with “human face” is additionally provided (like WIDER FACE [39]). As shown in Figure 1, many faces in the original training data are not labeled, similarly, the instances belonging to the original categories (e.g., “person”) are not annotated in the new dataset.

Figure 1: A unified detector is expected to detect all categories (the 80 classes in COCO and “human face”) while only separately labeled training sets are provided. Among the simple ways, relabeling is extremely costly and plain joint training leads to severe biases and poor performance. Best seen in color.
instances (e.g., “human face”) will be regarded as negative (background) samples, resulting in conflicts and false gradients during training. It is difficult for the detector trained in the plain way to detect categories which are not labeled in corresponding training data because of overfitting the false negatives. For example, as shown in Figure 1, the model trained on the combined training set of COCO and WIDER FACE cannot detect “human face” in COCO evaluation set or “person” in WIDER FACE evaluation set.

Can we obtain a unified detector that can detect all categories well while avoiding relabeling manually and alleviating conflicts during training? To this end, we propose a solution that mainly concerns three aspects. First, a conflict-free loss is proposed to pursue a usable base detector. It is carefully designed to avoid two possible conflicts during joint learning: (i) samples assigned as negatives in one dataset may be actually positives belonging to unlabeled instances with classes from the other datasets. (ii) samples assigned as positives may be more suitable to be labeled as categories of the other datasets. Second, we attempt to mine high-quality unlabeled ground-truth with the base detector. We employ the Monte Carlo Dropout to obtain localization confidence, which is combined with the classification confidence to unearth pseudo labels with more accurate bounding boxes. Third, to achieve a stronger detector, we explore several retraining strategies with the mined pseudo annotations. We realize that although a lot of unlabeled instances are mined, there still hide many latent ground-truth. Therefore, a conflict-free method cannot be removed in the retraining stage. However, simply copying the training strategy from the first phase results in insufficient negative information, as only positive samples are added by pseudo annotations during retraining. Consequently, we employ an overlap-based negatives weighting strategy to utilize negative samples modestly in the retraining process.

To sum up, the main contributions of this paper are listed as follows: (i) We present a general solution for training detectors based on only the original datasets and incrementally labeled new datasets, which is urgently required in realistic applications. (ii) A conflict-free loss is proposed to adapt to this problem. (iii) Better unlabeled ground-truth mining methods and retraining strategies with pseudo annotations are explored. (iv) Extensive experiments are conducted to demonstrate the effectiveness of our proposed solution as compared with state-of-the-art approaches. It is worth noting that the entire pipeline does not introduce any additional manual labeling or modification of network structures.

2. Related Work

Object Detection. Object detection has made significant progress in the past decade. Anchor-based detectors (R-CNN serials [11, 10, 32], SSD [24], YOLO [31], RetinaNet [22], etc.) and anchor-free detectors (DenseBox [17], CornerNet [21], FCOS [35], etc.) mainly focus on efficiency or performance enhancement based on sufficient and complete training data. Object detection in more complex and realistic scenarios has also attracted much attention recently, such as Semi-Supervised Object Detection (SSOD), Open-Set Object Detection (OSOD), Multi-Dataset Object Detection (MDOD) and Continual Object Detection (COD).

Semi-Supervised Object Detection. SSOD aims to learn detectors based on few labeled images and a significant number of unlabeled images. CSD [18] and FocalMix [36] utilize consistency regularization to make full use of the unlabeled data. Self-training and strong data augmentations are employed in STAC [34] to enhance the detector for SSOD. Although the issue of incomplete training data is also involved in the problem discussed in this paper, the main difference is that no extra images are introduced during training except the available partially-labeled datasets in our problem.

Open-Set Object Detection. OSOD [5, 13, 2, 27] refers to the issue that objects from new domain or new classes that are not seen in the training data may be encountered in the test data, which brings about more false positives compared to the closed-set. In this work, there probably exists non-ignorable domain gap between the new annotated datasets and the original datasets. Methods for OSOD inspire us to mine higher-quality pseudo annotations with less false positives based on uncertainty estimation.

Multi-Dataset Object Detection. MDOD tries to train a single detector on multiple datasets, which is as the same purpose as in this work. In [41] and [30], a pseudo labeling approach is exploited. Dataset-aware focal loss is proposed in [40] to avoid conflicts in the multi-dataset training. Compared to these methods, we present a more powerful pipeline to deal with this problem, which concentrates on three main questions: how to obtain a better base detector? how to mine more accurate pseudo annotations? and how to make better use of pseudo labels?

Continual Object Detection. COD, which aims at a lifelong learning detection system, is also related to our work. In [33], [14] and [29], they make great efforts to learn detectors incrementally based on only new training images with new classes, i.e., the original training data is not available when new classes arriving. In this setting, the main challenge is the catastrophic forgetting [25, 8] for old classes due to the lack of original datasets. While, in this paper, we focus on a more immediate and realistic situation in which the original training data is still available. Although it relaxes the constraints, we show that it is still a challenging problem which is often encountered in real world applications and needed to be addressed urgently.
3. Category-extended Object Detector

In this section, we introduce our solution for training category-extended detectors. We focus on three aspects to obtain a unified and powerful detector based on the incomplete datasets. First, we propose the conflict-free loss, which attempts to take full advantage of the exact information and avoid ambiguous one. Second, we design an unlabeled ground-truth mining process in which classification confidence and localization confidence based mining strategies are explored for object detection. Third, we explore better retraining strategies to obtain a stronger detector with the mined pseudo annotations. Figure 2 illustrates the diagram of our approach.

For brevity of presentation and without loss of generality, we assume that there are original dataset \( D_o \) (denoted by categories \( C_o \), images \( I_o \) and ground-truth \( G_o \)) and newly-added dataset \( D_n \) (denoted by categories \( C_n \), images \( I_n \) and ground-truth \( G_n \)), with different label spaces (i.e., \( C_o \cap C_n = \emptyset, C_o \) and \( C_n \) do not include the special category “background”). We aim to train a unified object detector on both \( D_o \) and \( D_n \). The overall loss function can be formulated as a weighted sum of the classification loss \( L_{cls} \) and the localization loss \( L_{loc} \):

\[
L([p_i], [t_i]; [p_i^*], [t_i^*]) = L_{cls}([p_i], [p_i^*]) + L_{loc}([t_i], [t_i^*]),
\]

where \( i \) is the index of an anchor, \( t_i \) is a 4-dimensional vector representing the parameterized coordinates of the predicted bounding box, and \( t_i^* \) is that of the ground-truth box associated with a positive anchor. The \( |C_o \cup C_n| \)-dimensional vector \( p_i^* \) is the ground-truth label for anchor \( i \). If anchor \( i \) matches with a ground-truth box in \( G_o \) or \( G_n \) of category \( c \), \( p_i^* \) is denoted as a one-hot vector with only \( p_i^{*,c} = 1 \). If it does not match any box in \( G_o \cup G_n \), \( p_i^* \) will be set to \( 0 \). \( p_i \) is the predicted probability vector of anchor \( i \).

### 3.1. Conflict-Free Loss

As the localization loss is activated only for positive anchors \( i \in \text{Pos} \) and disabled otherwise \( i \in \text{Neg} \), there are few possible conflicts hidden in it, so that we keep the localization loss unchanged. However, the classification loss in the multi-dataset training must be carefully designed to avoid possible conflicts of positive and negative samples. In this subsection, we propose a BCE-based loss, called Conflict-Free Loss, which attempts to make full use of the definitely correct information and avoid using ambiguous supervisory information at the same time.

**Binary Cross-Entropy (BCE) Loss.** BCE loss views the classification task as a series of independent binary classification tasks, which is formulated as

\[
L_{cls}([p_i], [p_i^*]) = \frac{1}{N_{cls}} \sum_{i,c} w(i,c) \cdot l_{bce}(p_i^{c*}, p_i^{c*}),
\]

where \( l_{bce}(p_i^{c*}, p_i^{c*}) = -[p_i^{c*} \log(p_i) + (1-p_i^{c*}) \log(1-p_i^{c})] \), and \( w(i,c) = 1, \forall \) anchor \( i \in I_o \cup I_n, c \in C_o \cup C_n \).

**Conflict-Free Loss.** The plain BCE loss will be affected by false signals during joint training on multiple datasets. To avoid this, we propose the BCE-based Conflict-Free loss, in which two possible conflict origins are removed. First, a negative sample will not contribute to the loss of classes that are not labeled in its dataset. Second, we consider that the assigned positive samples may actually belong to categories of other datasets. Commonly, in one dataset, an anchor is assigned as positive for one specific category when the Intersection over Union (IoU) with the ground-truth is greater than a threshold (say, 0.5 typically). However, such loose restriction may ignore the possibility that the anchor is more suitable to be labeled as another category which is only annotated in other datasets. To avoid this situation, the conflict-free loss only concerns positive anchors with remarkably large overlaps with ground-truth (IoU \( \geq \tau_{strict} \)), the more strict threshold \( \tau_{strict} \) is set to 0.9 conservatively) to provide negative information to classes from the other datasets. The conflict-free loss is formulated as Eq. 2 with

\[
w(i,c) = \begin{cases} 
0, & (i \in \text{Neg} \cap i \in I_o \cap c \notin C_o) \cap (i \in \text{Pos} \cap f(i) < \tau_{strict} \cap i \in I_o \cap c \notin C_o), \\
1, & \text{otherwise},
\end{cases}
\]

where, \( * \) represents \( o \) or \( n \), \( f(i) \) represents the maximum IoU of anchor \( i \) with ground-truth.

**Discussions.** (i) Similar to BCE loss, cross-entropy (CE) loss is also widely used. With CE loss, label \( p^* \) and classification prediction \( p \) should have \( |C_o \cup C_n| + 1 \) categories (plus the “background”) and the prediction \( p \) is activated by “softmax” instead of “sigmoid”. However, in CE loss, all categories would affect each other. So that the incorrect supervisory information cannot be eliminated as conveniently
as in BCE loss. (ii) The conflict-free loss is a unified formula, which can be used directly or modified slightly by replacing \( l_{\text{bce}} \) in Eq. 2 with any BCE-based loss (e.g., Focal loss [22]). The dataset-aware loss in [40] is actually a special case of conflict-free loss with focal loss and without considering the second conflict origin discussed above. An illustration of these losses is provided in supplementary material.

3.2. Unlabeled Ground-Truth Mining

In the previous section, we aggregate available reliable information as much as possible from old and new datasets to construct a unified base detector. Even though, there still exist large quantity of unlabeled objects that are underutilized. Inspired by self-training, we attempt to mine unlabeled ground-truth with the help of the base detector.

**Classification Confidence (CC) Based.** Similar to classification tasks [3, 7, 42], unlabeled ground-truth can be mined based on the classification confidence. Coarse annotations for the classes that are not labeled in a dataset can be generated through test time inference using the base detector. The generating process relies on a score threshold strategy, which exploits the classification score to represent the detector’s confidence in the predicted results. We accept a predicted bounding box as a pseudo label for class \( c \) only if the classification score \( p \) satisfies: (i) \( p^c = \max(p^1, \cdots, p^{|C\cup\mathcal{C}_n|}) \) is larger than a predefined threshold \( \eta \); (ii) class \( c \) is not labeled in the dataset originally. Generally speaking, only the predictions with enough high confidence will be selected as pseudo annotations to complete ground-truth for the unlabeled classes.

**Classification and Localization Confidence (CLC) Based.** Unlike classification tasks, object detection needs to predict the localization of objects, so that accurate localization annotations lead to better detectors. However, previous works [19, 16, 37] have proven that the classification score cannot be interpreted as the localization confidence since a predicted bounding box with high classification confidence may be localized inaccurately. Such inaccurate pseudo annotations will prevent detectors from getting better performance. In pursuit of high-quality pseudo annotations, we integrate localization confidence into the mining process.

Monte Carlo Dropout-based Bayesian neural networks have shown great potential in measuring model uncertainty [9, 20, 27, 26]. Inspired by this concept, during inference phase, we perform \( T \) forward passes of the base detector with activated dropout layers as depicted in Figure 3. Notice that since we introduce dropout operation only on the last layer of the classification and localization head while all the previous layers are identical and the computations are shared, the extra inference time during \( T \) forward passes is ignorable. Then, for each image, we perform clustering on the bounding boxes of each class, resulting a set of clusters. Each cluster \( O_i \) is made up of several bounding boxes with their respective classification score, defined as

\[
O_i = \{(t_{ij}, p_{ij}) \mid j = 1, 2, \cdots, |O_i| \leq T\},
\]

\[
s.t. \text{IoU}(t_{ij_1}, t_{ij_2}) \geq \tau_{\text{nms}}, \forall t_{ij_1}, t_{ij_2} \in O_i,
\]

\[
\text{argmax}_{r \in O_i \cup C_n} \text{IoU}(t_{ij_1}, t_{ij_2}, \forall p_{ij_1}, p_{ij_2} \in O_i,
\]

where, the threshold for clustering is the same as the NMS IoU threshold \( \tau_{\text{nms}} \) (0.5 typically). Then the cluster \( O_i \) is represented by a triplet \( (\mathbf{t}_i, \mathbf{p}_i, \mathbf{e}_i) \), where the integrated bounding box \( \mathbf{t}_i \) and classification confidence \( \mathbf{p}_i \) can be calculated by

\[
\mathbf{t}_i = \frac{1}{|O_i|} \sum_j t_{ij}, \quad \mathbf{p}_i = \frac{1}{|O_i|} \sum_j \max(p^1_{ij}, \cdots, p^{|C\cup\mathcal{C}_n|}_{ij}),
\]

More importantly, the localization confidence is given by

\[
\mathbf{e}_i = \frac{1 + \mathbb{1}_{|O_i| \geq T}}{2|O_i|^2} \sum_{j_1, j_2} \text{IoU}(t_{ij_1}, t_{ij_2}.
\]

Finally, as shown in Figure 3, we obtain the detection confidence \( \mathbf{d}_i = \mathbf{p}_i \times \mathbf{e}_i \). It considers both classification confidence and localization confidence, which better reflects the quality of predictions. The compositive detection confidence instead of the classification confidence is employed in the CLC-based unlabeled ground-truth mining. Other operations are similar to those in the CC-based mining method.

3.3. Retraining with Pseudo Annotations

After acquiring the pseudo annotations, we try to train a more powerful unified detector with the ground-truth and pseudo annotations together. Several strategies of utilizing the pseudo annotations may be considered as follows.

**As fully labeled.** The most straightforward way is viewing the combined dataset supplemented with pseudo annotations as fully labeled [34]. Then the detector can be trained in a normal way with common classification losses. But even though, there may still exist considerable instances that are difficult to be discovered by the base detector. Hence, there is a risk that the inaccurate supervisory information may be introduced.
Positives only. We can also retrain the detector with the conflict-free loss again. Compared to the first phase, the negative samples still do not contribute to the loss of classes from different datasets, only positive samples are supplemented by the pseudo annotations. So that the imbalance between positives and negatives may be intensified in this way, leading to a biased classification.

Positives & safe negatives. A solution to alleviate the imbalance problem mentioned above is to introduce some “safe” negatives as described in [41]. In addition to the threshold $\eta$ used to mine high-quality pseudo annotations, a lower threshold $\eta'$ is introduced to get high-recall pseudo annotations. Under the combination of ground-truth and high-quality pseudo annotations, some negatives are actually “unsafe” negatives (UnsafeNeg) which can match boxes from high-recall pseudo annotations. Anchors that do not match any ground-truth or high-recall pseudo annotations are considered to be “safe” negatives. “Safe” negatives can contribute to the loss of classes from different datasets while “unsafe” negatives are still discarded. This strategy can be formulated as training with Eq. 2 and

$$w(i, c) = \begin{cases} 0, & i \in \text{UnsafeNeg} & i \in I, & c \notin C_e, \\ 1, & \text{otherwise.} \end{cases} \quad (7)$$

Positives & weighted negatives. A soft overlap-based sample weighting approach [38] is introduced to reweight the negatives’ contribution to the loss of classes from different datasets. The main hypothesis is that negatives which have relatively large overlaps (like 0.3 IoU, 0.4 IoU) with the existing ground-truth boxes are probably not related to any unlabeled instances. Based on this, the Gompertz function $Gom(x) = ae^{-be^{-cx}}$ is employed to describe the relationship between overlap and weight, where $a$ is an asymptote, $b$ sets the displacement along the x-axis, $c$ sets the growth rate. We plot this function in Figure 4 with $a = 1$, $b = 10,000$, and $c = 25$. The negatives’ contribution to the loss of classes from different datasets is assigned with high weight only when they have large overlap with annotated boxes. Other negatives are still given very low weights. This strategy is formulated as training with Eq. 2 and

$$w(i, c) = \begin{cases} Gom(f(i)), & (i \in \text{Neg} & i \in I, & c \notin C_e) \mid (i \in \text{Pos} & f(i) < \tau_{strict} & i \in I, & c \notin C_e), \\ 1, & \text{otherwise}. \end{cases} \quad (8)$$

4. Experiments

4.1. Experimental Settings

Datasets. We conduct experiments on several widely used object detection datasets with different settings. The challenging MS COCO [23] (2017) consists of 118,287 training images (COCO-train), 5,000 validation images (COCO-val) and 40,670 test images with 80 categories. PASCAL VOC [6] (2007) has 9,963 images (50% for training/validation (VOC-train) and 50% for testing (VOC-test)) with 20 categories. WIDER FACE [39] is comprised of 32,203 images (training 40% (FACE-train), validation 10% (FACE-val) and test 50%) with one category “human face”. To evaluate the unified model in some settings, we also adopt the relabeled evaluation sets in [41]: VOC-subtest (a subset of VOC-test) contains 500 images with annotations for the 80 categories in COCO; the FACE-subval (a subset of FACE-val) consists of 500 images with annotations for the “human face” in COCO and the “person” in COCO; COCO-subval (a subset of COCO-val) has 500 images with annotations for the 80 categories in COCO and the “human face” in FACE.

Metrics. To evaluate the performance, we use the standard metrics for object detection, like AP (averaged AP at IoUs from 0.5 to 0.95 with an interval of 0.05), AP$_{50}$ (AP at IoU threshold 0.5), AP$_{75}$ (AP at IoU threshold 0.75).

Implementations details. For fair comparisons, all experiments are conducted on RetinaNet [22] with ImageNet-pretrained ResNet-50 [15] as the backbone. Models are implemented with PyTorch [28] and MMDetection [4]. Details are provided in supplementary material.

4.2. Ablation Study

We first analyze the effect of each component in our proposed pipeline. To simulate the scenario discussed in this work, we split COCO-train into two sets. One set is named COCO75-train, in which only annotations of 75 categories are retained, while annotations related to other 5 categories are removed; another set, named COCO5-train, contains only annotations of the 5 categories. COCO75-train and COCO5-train are regarded as original dataset $D_o$ and new dataset $D_n$, respectively. In COCO75-train, there exists considerable unlabeled objects of the remaining 5 categories, similar in COCO5-train. The statistics of the two sets are summarized in Table 1. We evaluate the methods on COCO-val and report the results for the 75 classes and the 5 classes respectively in Table 2.

Conflict-free loss. As shown in Table 2, the performance of the detectors trained separately on COCO75-train and COCO5-train are 36.3% AP and 20.2% AP respectively. The performance of the detector trained on the combined
Table 1: The statistics of the original dataset \( D_o \) and the new dataset \( D_n \) used in our experiments. More detailed information is provided in supplementary material.

|     | \( D_o \) | \( D_n \) | \( D_o \) | \( D_n \) | \( D_o \) | \( D_n \) | \( D_o \) | \( D_n \) |
|-----|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| #categories | COCO75-train | COCO5-train | COCO79-train | VOC-train | COCO60-train | COCO-train | FACE-train |
| #images | 75 | 5 | 79 | 1 | 118,287 | 5,011 | 118,287 | 12,880 |
| #annotations | 100,543 | 22,976 | 506,916 | 39,769 | 860,001 | 159,424 |
| #miss annotations | 129,089 | 106,526 | 222,696 | 90,620 | – | – | – | – |

Table 2: Performance of the detector trained on COCO75-train (\( D_o \)) and COCO5-train (\( D_n \)). We report the results of the 75 original classes and the 5 new classes on COCO-val, respectively. \( G_p \) stands for the pseudo annotations.

| Data | Method | 75 original classes | 5 new classes |
|------|--------|---------------------|--------------|
| \( D_o \) | Plain | 36.3 | 35.8 | 36.6 | 36.4 | 36.6 |
| \( D_n \) | Plain | 55.4 | 55.5 | 55.3 | 55.5 | 55.4 |
| fully labeled | Plain | 38.6 | 38.7 | 38.8 | 38.7 | 39.0 |
| \( D_o + D_n \) | Plain | 20.6 | 23.5 | 22.7 | 22.3 |
| Dataset-Aware [40] | 36.3 | 35.8 | 36.4 | 36.5 |
| Conflict-Free | 36.7 | 36.6 | 36.6 | 36.6 |
| \( D_o + D_n + G_p \) | CC + as fully labeled [34] | 36.6 | 36.6 | 36.7 | 36.6 |
| CC + positives only | 36.4 | 36.4 | 36.4 | 36.4 |
| CC + positives & safe negatives [41] | 36.5 | 36.5 | 36.5 | 36.5 |
| CC + positives & weighted negatives | 36.7 | 36.7 | 36.7 | 36.7 |
| CLC + positives & weighted negatives | 36.9 | 36.9 | 36.9 | 36.9 |

Unlabeled ground-truth mining. We investigate the performance of the CC-based and the CLC-based methods for unlabeled ground-truth mining. We expect more accurate bounding boxes can be mined by our CC-based mining method. To check the localization quality of bounding boxes, we count the true positives and the false positives with high overlap metric (\( \text{IoU} \geq 0.75 \)) in the predicted results whose classification confidence is larger than 0.5. The distribution of localization confidence across true positives and false positives is shown in Figure 5 (left). We find the overwhelming majority of bounding boxes with low localization confidence are false negatives, which shows the localization confidence can reflect the quality of bounding boxes to some extent. Besides, we show that the number of false positives can be suppressed with the collaboration of localization confidence as shown in Figure 5 (right). Table 2 also demonstrates that the pseudo annotations mined by the CLC-based method bring an improvement compared to the CC-based method.

Retraining strategies with pseudo annotations. We study the four strategies described in Sec. 3.3 for retraining with pseudo annotations. All pseudo annotations are generated with the plain CC-based mining strategy in these experiments. Table 2 compares the results of the four methods. Retraining by treating the training data with pseudo annotations as fully labeled leads to improvement compared to the training set in a vanilla way drops to 35.8% AP for the 75 classes, which is mainly caused by the conflicts during joint training; and improves to 20.6% AP for the 5 classes, which reveals that more training data leads to better feature representation and higher detection performance. A significant gap appears between the models trained on the incomplete data and the fully labeled data (35.8% vs. 37.1% AP and 20.6% vs. 30.6% AP for the 75 classes and 5 classes respectively). The detector trained with dataset-aware loss [40] outperforms the plain method. Furthermore, our conflict-free loss achieves better results (23.5% AP for the 5 classes), which mainly due to that the conflict-free loss can avoid two possible conflicts simultaneously as discussed in Sec. 3.1.
results of the first phase (0.1%AP and 0.5%AP gains for the 75 classes and the 5 classes, respectively), which demonstrates the effectiveness of the unlabeled ground-truth mining. The detector simply retrained with the conflict-free loss again (positives only) achieves better performance for the 5 classes but worse results for the 75 classes. Using positives and safe negatives in the retraining process reaches 24.5% AP for the 5 classes. Finally, the detector retrained with the weighted negatives achieves the best results (24.7% AP for 5 classes). These results imply that there are still many unlabeled ground-truth in the training data, so that the conflict-free method cannot be removed in the retraining process. Whereas, if we use the same training method as the first phase, only positives would be added, causing insufficient negative information in retraining. Therefore, it is important to employ suitable negative samples to achieve balance between positives and negatives. Our empirical experiments show that using the weighted negatives is a better choice for the purpose.

4.3. More Comparisons

To further evaluate the effectiveness of our method, more experiments on different dataset settings (refer to Table 1) are conducted and analyzed as follows.

COCO79-train and COCO1-train. We perform another split on COCO-train: the original dataset COCO79-train contains 79 classes, and the new dataset COCO1-train has only one category. We report the performance of the 79 classes and the one class on COCO-val respectively in Table 3. The conflict-free loss also leads to a better detector than the plain method. After retraining with the pseudo annotations (generated by the CLC-based mining method) and the weighted negatives, the performance of the 79 classes marginally decreases by 0.3% AP, while achieves a remarkable gain of 3.1% AP for the new class and outperforms other retraining strategies proposed in [34] and [41].

COCO60-train and VOC-train. In these experiments, we regard VOC-train as new dataset and COCO60-train as original dataset, which is generated by removing the annotations of the 20 VOC categories from COCO-train. We report the performance on VOC-subtest and COCO-subval in Table 4. We also show the results of the 20 VOC categories on COCO-val. Consistent with the above results on

Table 3: Performance of the detector trained on COCO79-train ($D_o$) and COCO1-train ($D_n$). We report the results of the 79 original classes and the 1 new class on COCO-val, respectively.

| Data                  | Method                              | 79 original classes | I new class |
|-----------------------|-------------------------------------|---------------------|-------------|
|                       |                                     | AP | AP50 | AP75 | AP | AP50 | AP75 |
| fully labeled         | Plain                               | 36.5 | 55.5 | 38.9 | 51.2 | 79.6 | 54.2 |
| $D_o + D_n$           | Plain                               | 35.4 | 54.4 | 37.7 | 37.9 | 69.2 | 36.5 |
| $D_o + D_n$           | Conflict-Free                       | 36.5 | **55.5** | **39.0** | 41.6 | 72.4 | 41.4 |
| $D_o + D_n + G_p$     | CC + as fully labeled [34]          | 36.2 | 55.1 | 38.4 | 42.9 | 71.1 | 44.4 |
| $D_o + D_n + G_p$     | CC + positives only                 | 36.1 | 54.8 | 38.3 | 44.4 | 74.4 | 45.6 |
| $D_o + D_n + G_p$     | CC + positives & safe negatives [41]| 36.1 | 55.0 | 38.2 | 43.7 | 72.9 | 45.0 |
| $D_o + D_n + G_p$     | CC + positives & weighted negatives | 36.0 | 54.8 | 38.1 | 44.6 | 74.2 | 46.0 |
| $D_o + D_n + G_p$     | CLC + positives & weighted negatives| 36.2 | 55.0 | 38.5 | **44.7** | **74.6** | **46.3** |

Table 4: Performance of the detector trained on COCO60-train ($D_o$) and VOC-train ($D_n$). We report the results of all 80 classes on VOC-subtest and COCO-subval, and the 20 VOC classes on COCO-val.

| Data                  | Method                              | VOC-subtest | COCO-subval | COCO-val |
|-----------------------|-------------------------------------|-------------|-------------|----------|
|                       |                                     | all 80 classes | all 80 classes | 20 new classes |
|                       |                                     | AP50 | AP75 | AP50 | AP75 | AP50 | AP75 |
| $D_o + D_n$           | Plain                               | 43.0 | 28.0 | 42.6 | 27.2 | 4.1 | 2.7 |
| $D_o + D_n$           | Conflict-Free                       | 46.8 | 30.1 | 50.3 | 30.8 | 32.3 | 18.6 |
| $D_o + D_n + G_p$     | CC + as fully labeled [34]          | 48.0 | 31.5 | 48.7 | 29.9 | 27.5 | 18.6 |
| $D_o + D_n + G_p$     | CC + positives only                 | **49.4** | 31.7 | 49.9 | 30.8 | 34.1 | 20.4 |
| $D_o + D_n + G_p$     | CC + positives & safe negatives [41]| 49.0 | **33.3** | 49.9 | 31.5 | 34.3 | 21.7 |
| $D_o + D_n + G_p$     | CC + positives & weighted negatives | 47.5 | 31.4 | 51.6 | 31.6 | **42.7** | 24.9 |
| $D_o + D_n + G_p$     | CLC + positives & weighted negatives| 49.0 | 33.0 | **52.2** | **31.7** | 42.6 | **25.4** |
Table 5: Performance of the detector trained on COCO-train ($D_o$) and FACE-train ($D_n$). We report the results of categories “person” and “human face”. Blue indicates the results of categories that we should pay more attention to.

| Data               | Method                          | COCO-subval | FACE-subval |
|--------------------|---------------------------------|-------------|-------------|
|                    |                                 | person  | human face | person  | human face |
| $D_o + D_n$        | Plain                           | 71.1  | 40.8 | 0.9  | 6.7 | 3.7 | 54.3 | 28.0 |
| $D_o + D_n$        | Conflict-Free                   | 71.9  | **43.2** | 49.9 | 20.9 | 55.6 | 33.9 | 53.8 | 28.8 |
| $D_o + D_n + G_{pp}$ | CC + as fully labeled [34] | **72.7** | 42.6 | 49.0 | 22.6 | 51.5 | 31.8 | 53.8 | 28.9 |
| $D_o + D_n + G_{pp}$ | CC + positives only            | 71.2 | 42.1 | 53.9 | 22.8 | 56.4 | **34.7** | 53.7 | 28.8 |
| $D_o + D_n + G_{pp}$ | CC + positives & safe negatives [41] | 72.4 | 42.3 | 50.7 | 22.4 | 52.9 | 31.7 | 54.2 | **29.0** |
| $D_o + D_n + G_{pp}$ | CC + positives & weighted negatives | 71.4 | 42.8 | **54.0** | 22.5 | 58.3 | **34.7** | 53.1 | 28.7 |
| $D_o + D_n + G_{pp}$ | CLC + positives & weighted negatives | 72.3 | 43.1 | 53.9 | **23.7** | 58.5 | **34.7** | 53.2 | **29.0** |

Figure 6: Detection results of the plain method (top) and our method (bottom) on COCO-val (the first two columns) and FACE-val (the last two columns). Red bounding boxes indicate objects belonging to the 80 categories in COCO, while green ones corresponding to the category “human face” in WIDER FACE. Best seen in color.

other datasets, the proposed method exhibits much better performance. Especially for such shifted-domain scenario, CLC-based mining strategy leads to a more robust detector.

**COCO-train and FACE-train.** We also train unified detectors on COCO-train (as original dataset) and FACE-train (as new dataset). “human face” are not annotated in COCO-train and the 80 classes (e.g., “person”) in COCO-train are also not labeled in FACE-train. The performance on COCO-subval and FACE-subval is reported in Table 5. We focus on two representative categories: “person” and “human face”. Similar to the experiments on COCO60-train and VOC-train, directly training a detector on COCO-train and FACE-train results in a biased model and very poor performance, especially for the classes in a domain that are not labeled in the training data (e.g., “human face” on COCO-subval only gets 3.4% AP$_{50}$ and “person” on FACE-subval only obtains 6.7% AP$_{50}$). The proposed method can lead to a stronger unified detector for both the original classes and the new classes in generally and significantly outperforms the approaches proposed in [34] and [41].

**Visualization results.** Figure 6 illustrates some visualization results of the detectors trained on COCO-train and FACE-train. More results are available in supplementary material. Due to the conflicts during joint training and the domain gap between the two sets, the detector trained in a vanilla way only can detect “person” in COCO-val and “human face” in FACE-val (i.e., categories which are labeled in corresponding training data). Assisted by our approach, the detector can detect all categories on both evaluation sets. Both qualitative and quantitative results demonstrate the consistent effectiveness of our training pipeline.

5. Conclusion

In this paper, we aim at training a category-extended object detector without relabeling or conflicts. To achieve this goal, a powerful pipeline is proposed which consists of three parts. First, to obtain a usable base detector, the conflict-free loss is proposed to eliminate two possible conflicts during joint learning. Second, we employ a Monte Carlo Dropout-based localization confidence estima-
dition method to unearth more accurate pseudo annotations. Third, we explore several strategies for retraining with the pseudo annotations, and empirically exhibit that employing the positive samples supplemented with pseudo labels and weighted negatives during retraining leads to better performance. The experimental results on multiple settings demonstrate that the proposed method achieves a feasible category-extended object detector and outperforms other state-of-the-art approaches.

References

[1] Samet Akcay, Mikolaj E Kundegorski, Chris G Willcocks, and Toby P Breckon. Using deep convolutional neural network architectures for object classification and detection within x-ray baggage security imagery. *IEEE transactions on information forensics and security*, 13(9):2203–2215, 2018.

[2] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 384–400, 2018.

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5049–5059, 2019.

[4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiaru Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[5] Akshay Dhamija, Manuel Gunther, Jonathan Ventura, and Terrance Boult. The overlooked elephant of object detection: Open set. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 1021–1030, 2020.

[6] Mark Everingham, Van Gool, Christopher WI, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[7] Geoffrey French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. *arXiv preprint arXiv:1706.05208*, 2017.

[8] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

[9] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International conference on machine learning*, pages 1050–1059, 2016.

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

[11] 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.

[12] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocis, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020.

[13] David Hall, Feras Dayoub, John Skinner, Haoyang Zhang, Dimity Miller, Peter Corke, Gustavo Carneiro, Anelia Angelova, and Niko Sünderrauf. Probabilistic object detection: Definition and evaluation. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 1031–1040, 2020.

[14] Yu Hao, Yanwei Fu, Yu-Gang Jiang, and Qian Tian. An end-to-end architecture for class-incremental object detection with knowledge distillation. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2019.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[16] Yihui He, Chenchen Zhu, Jianren Wang, Marios Savvides, and Xiangyu Zhang. Bounding box regression with uncertainty for accurate object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2888–2897, 2019.

[17] Lichao Huang, Yi Yang, Yafeng Deng, and Yinan Yu. Densebox: Unifying landmark localization with end to end object detection. *arXiv preprint arXiv:1509.04874*, 2015.

[18] Jisoo Jeong, Seungeui Lee, Jeesoo Kim, and Nojun Kwak. Consistency-based semi-supervised learning for object detection. In *Advances in neural information processing systems*, pages 10759–10768, 2019.

[19] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of localization confidence for accurate object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 784–799, 2018.

[20] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584, 2017.

[21] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 734–750, 2018.

[22] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Jian Sun. Focal loss for dense object detection. In *Proceedings of the European Conference on Computer Vision*, pages 2980–2988, 2017.

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

[24] 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.

[25] Michael McClooskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning
problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989. 2

[26] Dimity Miller, Feras Dayoub, Michael Milford, and Niko Sünderhauf. Evaluating merging strategies for sampling-based uncertainty techniques in object detection. In 2019 *International Conference on Robotics and Automation (ICRA)*, pages 2348–2354. IEEE, 2019. 4

[27] Dimity Miller, Lachlan Nicholson, Feras Dayoub, and Niko Sünderhauf. Dropout sampling for robust object detection in open-set conditions. In 2018 *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7. IEEE, 2018. 2, 4

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019. 5, 11

[29] Juan-Manuel Perez-Rua, Xiatian Zhu, Timothy M Hospedales, and Tao Xiang. Incremental few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13846–13855, 2020. 2

[30] Alexandre Rame, Emilien Garreau, Hedi Ben-Younes, and Charles Ollion. Omnia faster r-cnn: Detection in the wild through dataset merging and soft distillation. *arXiv preprint arXiv:1812.02611*, 2018. 2

[31] 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, 2

[32] 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, 2

[33] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. Incremental learning of object detectors without catastrophic forgetting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3400–3409, 2017. 2

[34] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020. 2, 4, 6, 7, 8

[35] Zhe Wu, Navaneeth Bodla, Bharat Singh, Mahyar Najibi, Rama Chellappa, and Larry S Davis. Soft sampling for robust object detection. *arXiv preprint arXiv:1806.06986*, 2018. 5

[36] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5982–5991, 2019. 4
A. Detailed information of datasets.

The categories of the training datasets used in our experiments are presented in Table 6.

B. Implementation details.

For fair comparisons, all experiments are conducted on RetinaNet [22] with ImageNet-pretrained ResNet-50 [15] as the backbone. The models are implemented with PyTorch [28] and MMDetection [4]. In all experiments, input images are resized to 1333 × 800, without changing the aspect ratio. The models are trained using SGD over 8 GPUs with 2 images per GPU with 0.9 momentum, 0.0001 weight decay. We train 12 epochs in total with an initial learning rate of 0.02, and decrease the learning rate by 0.1 at epoch 8 and 11. For the conflict-free loss, the more strict threshold \( \tau_{\text{strict}} \) is set to 0.9 conservatively as described in Sec. 3.1 in all experimental settings. For the CC-based unlabeled ground-truth mining method, the confidence threshold \( \eta \) is set to 0.5 in all experimental settings for best overall performance. For the CLC-based unlabeled ground-truth mining method, the number of forward passes \( T \) is set to 20 for all experimental settings; the confidence threshold \( \eta \) is set to 0.525 in the experiment on COCO75-train and COCO5-train, and 0.5 in other experimental settings for best overall performance. For the retraining strategy “Positives & safe negatives”, the lower confidence threshold \( \eta' \) is set to 0.1 to obtain high-recall pseudo annotations in all experimental settings. For the retraining strategy “Positives & weighted negatives”, we set \( \tilde{a} = 1 \), \( \tilde{b} = 10,000 \) and \( \tilde{c} = 25 \) as shown in Sec. 3.3 in all experimental settings. There is almost no change in hyper-parameters in different experimental settings, which shows that our pipeline is robust.

C. Illustration of the conflict-free loss.

An illustration of the plain binary cross-entropy loss and the conflict-free loss is presented in Figure 7. In the BCE-based Conflict-Free loss, two possible conflict origins are removed. First, a negative sample will not contribute to the loss of classes that are not labeled in its dataset. Second, the conflict-free loss only concerns positive anchors with remarkably large overlaps with ground-truth to provide negative information to classes from the other datasets.

D. More visualization results.

Figure 8 and Figure 10 illustrate more visualization results of the detectors trained on COCO-train and FACE-train. Figure 9 and Figure 11 illustrate the visualization results of the detectors trained on COCO60-train and VOC-train. These results also show that our method achieves more powerful and robust detectors under the limited data.
Figure 7: Illustration of the conflict-free loss. We give an example on the original dataset, the operation is similar on the new dataset. Best seen in color.
Figure 8: Detection results of the plain method (left) and our method (right) on COCO-val (top) and FACE-val (bottom). Red bounding boxes indicate objects belonging to the 80 categories in COCO, while green ones corresponding to the category “human face” in WIDER FACE. Best seen in color.
Figure 9: Detection results of the plain method (left) and our method (right) on COCO-val (top) and VOC-test (bottom). Red bounding boxes indicate objects belonging to the 60 categories in COCO60-train, while green ones corresponding to the 20 VOC categories. Best seen in color.
Figure 10: Detection results on COCO-val (the first two columns) and FACE-val (the last two columns). Red bounding boxes indicate objects belonging to the 80 categories in COCO, while green ones corresponding to the category “human face” in WIDER FACE. (a) Plain, (b) Conflict-Free, (c) CC + as fully labeled, (d) CC + positives only, (e) CC + positives & safe negatives, (f) CC + positives & weighted negatives, (g) CLC + positives & weighted negatives. Best seen in color.
Figure 11: Detection results on COCO-val (the first two columns) and VOC-test (the last two columns). Red bounding boxes indicate objects belonging to the 60 categories in COCO60-train, while green ones corresponding to the 20 VOC categories. (a) Plain, (b) Conflict-Free, (c) CC + as fully labeled, (d) CC + positives only, (e) CC + positives & safe negatives, (f) CC + positives & weighted negatives, (g) CLC + positives & weighted negatives. Best seen in color.
Table 6: Categories of the training datasets.

| COCO75-train                                                                 | COCO5-train       |
|------------------------------------------------------------------------------|-------------------|
| person, bicycle, motorcycle, airplane, bus, train, truck, boat,               | car, bottle, cup, |
| stop sign, parking meter, bench, bird, cat, dog, horse, sheep, cow, elephant, | chair, book       |
| backpack, umbrella, handbag, tie, suitcase, frisbee, skis, snowboard,        |                   |
| baseball glove, skateboard, surfboard, tennis racket, wine glass, fork,       |                   |
| apple, sandwich, orange, broccoli, carrot, hot dog, pizza, donut, cake,      |                   |
| couch, potted plant, bed, dining table, toilet, tv, laptop, mouse,           |                   |
| remote, keyboard, oven, toaster, sink, refrigerator, clock,                  |                   |
| vase, scissors, teddy bear, hair drier, traffic light, fire hydrant, bear,    |                   |
| zebra, giraffe, baseball bat, bowl, banana, microwave, cell phone, kite,     |                   |
| knife, spoon, toothbrush, sports ball,                                       |                   |
| COCO79-train                                                                 | COCO1-train       |
| person                                                                       |                   |
| bicycle, car, motorcycle, airplane, bus, train, truck, boat, traffic light,  |                   |
| stop sign, parking meter, bench, bird, cat, dog, horse, sheep, cow, elephant,|                   |
| backpack, umbrella, handbag, tie, suitcase, frisbee, skis, snowboard,        |                   |
| baseball glove, skateboard, surfboard, tennis racket, bottle, wine glass,    |                   |
| cup, banana, apple, sandwich, orange, broccoli, carrot, hot dog, pizza,      |                   |
| donut, cake, potted plant, bed, dining table, toilet, tv, laptop, mouse,     |                   |
| remote, keyboard, oven, toaster, sink, refrigerator, book, clock,             |                   |
| vase, scissors, teddy bear, hair drier, fire hydrant, bear, zebra, giraffe,   |                   |
| kite, baseball bat, fork, knife, spoon, bowl, chair, couch, cell phone,      |                   |
| microwave, hair drier, toothbrush, sports ball,                               |                   |
| COCO60-train                                                                 | VOC-train         |
| person, bicycle, car, tv (tvmonitor)                                        |                   |
| train, boat, potted plant (pottedplant)                                      |                   |
| chair, couch (sofa), bird, cat,                                              |                   |
| bus, cow, bottle, airplane (aeroplane), dining table (diningtable),          |                   |
| sheep                                                                         |                   |
| motorcycle (motorbike), dog, horse,                                          |                   |
| tie, suitcase, frisbee, skateboard, surfboard, tennis racket,                |                   |
| sandwich, orange, broccoli, carrot, keybd,                                   |                   |
| COCO-train                                                                   | FACE-train        |
| person, bicycle, car, motorcycle, airplane, bus, train, truck, boat,         | human face        |
| stop sign, parking meter, bench, bird, cat, dog, horse, sheep, cow,           |                   |
| backpack, umbrella, handbag, tie, suitcase, frisbee, skis, snowboard,        |                   |
| baseball glove, skateboard, surfboard, tennis racket, bottle, wine glass,    |                   |
| banana, apple, sandwich, orange, broccoli, carrot, hot dog, pizza,           |                   |
| potted plant, bed, dining table, toilet, tv, laptop, mouse, remote,          |                   |
| traffic light, fire hydrant, elephant, bear, zebra, giraffe,                  |                   |
| sports ball, kite, baseball bat, cup, fork, knife, spoon, bowl, donut,      |                   |
| cake, chair, couch, keyboard, cell phone, microwave, oven, toaster, sink,    |                   |
| refrigerator, book, clock, vase, scissors, teddy bear, hair drier, toothbrush,|                   |