Abstract

This report introduces our solution for the ICCV 2021 Workshop SSLAD Track 1 - 2D Object Detection. The goal of Semi-Supervised Object Detection (SS-OD) is to train a detector that exploits large amounts of unlabeled data with only a few data labeled. Till now, related methods can be roughly divided into self-training based methods and consistency-regularization based methods. In this work, we combined the above mentioned methods and proposed a holistic framework named O2O (Offline to Online) for the SS-OD problem. We adopted Cascade R-CNN with the Swin-Transformer backbone as our detector for training on the SODA10M dataset. In virtue of the proposed SS-OD method, our best single model could reach an mAP of 81.35. With ensembles, we further improved the final mAP in the public leaderboard to 81.79 and achieved 2nd place in the SSLAD Track 1 challenge.

1. Introduction

The SSLAD 2D Object Detection challenge in ICCV 2021 is a 2D object detection task for autonomous driving. A large-scale dataset called SODA10M [5] with 10 million unlabeled images in total is provided. For labeled set, SODA10M annotates 5K images for training, 5K images for validation, and 10K images for testing. The main task of this track is to exploit the massive unlabeled data together with labeled images to improve the performance and generalize on the test set. A straightforward way to address this challenge is to adopt Semi-Supervised Object Detection (SS-OD) methods.

methods with a self-training patternWhile semi-supervised learning has been widely explored in image classification tasks [11, 13], few works have been focused on object detection due to the difficulty of localizing and regressing the location of each object on images. Recent SS-OD methods can be roughly divided into self-training based methods and consistency-regularization based methods. The self-training based methods [12] generate pseudo labels from a pre-trained model and then train the detector jointly with unlabeled and labeled data. We name those methods with a self-training pattern as offline methods, as the pseudo labels are fixed during the semi-supervised training process. On the other hand, consistency-regularization based methods [8, 15] dynamically generate pseudo labels or regularize the consistency of outputs with different data transformations. Most of them are in a Teacher-Student pattern with the teacher model updated via Exponential Moving Average (EMA) of student weights. Those methods are named Online methods as the predictions vary through iterations.

The different training patterns endue those methods with different characteristics. We argue that while pseudo labels generated by offline methods can be of high quality, the gain margin is limited with fixed predictions. For online methods, the model outputs gradually evolve through time if trained properly although the performance can be inferior in the early training stage. It’s validated that self-training based methods showed inferior performance than consistency-regularization based methods and the performance couldn’t improve with an increased amount of unlabeled images in experiments [5]. Therefore, we proposed a holistic SS-OD framework called O2O (Offline to Online) to combine the advantages of above mentioned methods. We train our model in the following steps: 1) train a baseline detector on labeled images, 2) first utilize the fixed pseudo labels generated on unlabeled images by the baseline detector to train the student detector, and then 3) switch to Teacher-Student training pattern after a period of iterations. Also, we can repeat the process a few times to further improve the performance as the work [14].

As in a 2D object detection task, it’s equally important to design a strong detector for training on limited labeled images so that it can generalize well on unseen unlabeled data. We adopted Cascade R-CNN [2] with Swin Transformer [9] as backbones. Swin Transformer recently refreshes the SOTA results of multiple computer vision tasks including
object detection and enables us to have a strong detection baseline as well as generate high-quality pseudo labels. To summarize our whole solution:

- **Strong Detector**: We adopted Cascade R-CNN with Swin Transformer backbones, which provides a strong baseline and generalizes well on unseen images.

- **O2O Framework**: We proposed an SS-OD framework named as O2O combining the advantages of self-training based methods as well as consistency-regularization based methods to improve the detection performance.

- **Auto Ensemble**: We adopted a search-based ensemble mechanism to aggregate predictions by different detectors automatically.

### 2.2D Object Detection

We use two-stage Cascade R-CNN [2] as our detector and adopted Swin Transformer [9] as the backbone feature extractor. The task of this challenge is to exploit massive unlabeled data with limited labeled images and we assume that the diversity of model architectures won’t contribute much to achieving an impressive performance.

#### Basic Architecture

The detector architecture we use is the same as that of the original paper. It should be noticed that when training on only labeled images to get a baseline detector, we add auxiliary mask heads like HTC [3] did to help improve the detection performance by learning hybrid tasks. The fed masks are forged by simply filling pixels within annotated bounding boxes and this design gives us a 0.75% boost in mAP. In semi-supervised training, the mask heads are removed to reduce GPU memory usage. Other improvements such as replacing the L1 loss with GIoU loss [10] also work well in our training.

#### Data Augmentation

With limited labeled images and complex driving scenes, overfitting is introduced unavoidably in this task. Except for standard augmentation strategies such as multi-scale training and flip, we also use more radical augmentations like color jitters and MixUp [16]. Those augmentations are helpful especially for adjusting to lighting changes in different scenes and detecting blocked objects. Other techniques like Copy-Paste [4] and Mosaic augmentations [11] couldn’t lead to stable improvements.

#### TTA and Multi-class NMS

We empirically adopted standard test-time augmentation strategies, including multiscales and image flip. The chosen scale ratios are 1.0, 1.1, 1.2, and thus the TTA results are the average of 6 predictions in total. In the post-processing stage of object detection, NMS or soft-NMS is commonly used to filter invalid bounding boxes. We’ve found in our experiments that direct use of soft-NMS increases the final mAP by a minor improvement. The performance of pedestrian, cyclist, and tricycle actually degrades with the use of soft-NMS while the mAP of remaining threes increase. After analyzing, we assume that soft-NMS could retrieve more miss detections for categories with a higher IoU threshold, which is 0.7 for car, truck, and tram. While soft-NMS would lead to more false detections for categories with a 0.5 IoU threshold, we merge the post-processing results of soft-NMS and NMS. This gives us a 0.3 ~ 0.4% boost in mAP.

### 3. O2O SS-OD Framework

Our SS-OD framework is shown in Figures [4] and we’ve divided the whole training process into three different parts.

#### 3.1. Stage-0: Baseline Training

This stage refers to training on labeled images to offer a good initialization for semi-supervised training as well as high-quality offline pseudo labels. In this stage, we add auxiliary mask heads to help learn latent knowledge and remove those heads in all other stages. For labeled data \(D_s = \{x_i^s, y_i^s\}_{i=1}^N\), we follow standard training schemes and the supervised loss consists of three parts:

\[
L = \sum_i L_{rpn}(x_i^s, y_i^s) + L_{roi}(x_i^s, y_i^s) + L_{mask}(x_i^s, y_i^s)
\]

The loss \(L_{rpn}\) and \(L_{roi}\) both include regression loss and classification loss. The trained weights \(\theta\) would be loaded for student model \(\theta_s\) and teacher model \(\theta_t\) in the next stage of training.

#### 3.2. Stage-1: O2O Semi-Supervised Training

To exploit large amounts of unlabeled images, we proposed O2O semi-supervised framework to combine offline and online methods. Offline pseudo labels generated by the baseline detector are first used to stabilize the model performance. As the training processes, the model could no longer learn useful knowledge from fixed pseudo labels and then we switch to a teacher-student pattern to continue training with the help of online pseudo labels.

#### Data Selection

Considering 10 million unlabeled images are available, we’ve tried different strategies to sample unlabeled data with a fixed amount to maximize the semi-supervised performance, such as sample uniformly from different annotated tags, but found out that random sampling is good enough for training. With collected data, we sample the labeled and unlabeled images separately to balance the ratio of those two in a single batch so that the training performance won’t be affected much by the errors in pseudo labels. Especially, the sampling ratio 1 : 1 is used, which means that the labeled set would be iterated much more times than the unlabeled set.

#### Offline Pseudo Labels

To get high-quality pseudo labels, we use TTA strategy mentioned in section 2 to boost the accuracy. To prevent the detrimental effects introduced
by noisy pseudo labels, it’s critical to set a proper confidence threshold $\sigma$ to filter false detections with low confidence, while keeping sufficient positive detections. We analyze the recalls and precisions for categories under different score thresholds as shown in Figure 2. The tendency of recalls and precisions become steady with the increase of the confidence threshold and we simply choose 0.5 as the threshold for offline pseudo labels as we find that further tuning the parameters only brings negligible improvements.

Online Pseudo Labels. When generating online pseudo labels, we adopt a teacher-student pattern where the teacher model is updated by the exponential moving average of the student model. The slowly updated teacher model $\theta_t$ ensures stabler and better detection performance than the student model $\theta_s$.

\[
\theta_t = \alpha \theta_t + (1 - \alpha) \theta_s \tag{2}
\]

As the false detections would easily be amplified through EMA updates, we set a higher threshold of 0.7 for filtering online pseudo labels to relieve the confirmation bias or error accumulation problem.

**Weak-Strong Augmentation.** It is common to use the weak-strong augmentation scheme in semi-supervised classification tasks [11]. In our proposed framework, we randomly apply weak and strong augmentations $A_{\text{rand}}$ on labeled images considering that the labeled data should stabilize the training process as well as avoid overfitting. For unlabeled data, the teacher model is fed with weakly augmented images to generate pseudo labels, and the student model is trained with strong augmentations $A_s$ for regularization purposes.

The final semi-supervised loss is the sum of the labeled loss and unlabeled loss. For unlabeled data $D_u = \{x_u^i\}_{i=1}^{N_u}$, the pseudo labels $\hat{y}_u^i$ would switch from fixed predictions generated offline by baseline detector and to online predictions made by the EMA teacher model after a period of iterations.

\[
L_s = \sum_i \mathcal{L}_{\text{rpn}}(A_{\text{rand}}(x_t^i), y_t^i) + \mathcal{L}_{\text{roi}}(A_{\text{rand}}(x_t^i), y_t^i) \tag{3}
\]

\[
L_u = \sum_i \mathcal{L}_{\text{rpn}}(A_s(x_u^i), \hat{y}_u^i) + \mathcal{L}_{\text{roi}}(A_s(x_u^i), \hat{y}_u^i) \tag{4}
\]

\[
L = L_s + L_u \tag{5}
\]

3.3. Stage-n: Iterative Self-Training

After the last stage of semi-supervised training, we would have a much better model than the baseline detec-
tor. Thus, it becomes possible to further improve the performance by putting back the trained semi-supervised model as the baseline detector and repeat the above process for more rounds. It should be noticed that after a round of semi-supervised training, the model becomes much more confident on unlabeled images and generate high-confidence predictions. This can be risky as the predefined threshold could no longer work especially for teacher-student training. As a result, we simply use the fixed pseudo labels generated by the trained model from the last stage instead of online predictions for fine-tuning.

4. Auto Ensemble

We design a two-stage auto ensemble scheme, in which the proposals of all models are fused and feed to ROI Heads respectively to produce final results. Considering the computational consumption required by semi-supervised training, we choose only 4 models differing slightly in training to a total of 24 predictions.

RPN Heads Ensemble. All predictions are simply concatenated together and the scores are lowered by soft-NMS while keeping as many positive detections as possible. For each, we retain the top 1000 boxes with the highest scores and feed them to the following ROI heads.

ROI Heads Ensemble. The predictions of each fed proposal in ROI heads would be weighted average into one single result. For every model, the weights of each category are first searched to find the best matches, which maximize the detection performance on the given validation set [7]. The ensemble process is presented as:

\[
(x, y) = \frac{w_1 \cdot (x_1, y_1) + w_2 \cdot (x_2, y_2) + \cdots + w_n \cdot (x_n, y_n)}{w_1 + w_2 + \cdots + w_n} \quad (6)
\]

\[
s = \left( \frac{w_1 \cdot s_1 + w_2 \cdot s_2 + \cdots + w_n \cdot s_n}{w_1 + w_2 + w_n} \right)^p \quad (7)
\]

where \((x, y)\) and \(s\) indicate the coordinates and confidence of the ensemble bounding box. The \(w_i\) is the average weight and \(p \in (0, 1]\) is used to control the confidence distribution, which helps retrieve more positive detections especially combined with soft-NMS. First, we would search for the post-processing method and IoU thresholds on the validation set. And then the simulated annealing method [6] is used to search for parameters \(w\) and \(p\).

5. Experiments

5.1. Implementation Details

Baseline Training. The baseline experiment is conducted on 8 NVIDIA V100 GPUs with a batch size of 16. In the training phase, we utilize multi-scale training and the image size ranges from \(960 \times 640\) to \(2880 \times 1920\). The SGD optimizer with an initial learning rate 0.03, momentum 0.9, and weight decay 0.0001 is used for training. The augmentation strategy includes mixup, cutmix, and random color jitters. The training epoch is set to 50 with the learning rate decayed by a factor of 0.1 at epochs 33 and 44.

The Cascade R-CNN detector uses Swin Transformer as the backbone with window size \(12 \times 12\), drop path rate 0.3, and ImageNet pre-trained weights for initialization. The baseline box head uses GIoU loss for regression and cross-entropy loss for classification. We keep the mask heads of HTC in baseline training and remove them in the following semi-supervised training for saving GPU memories.

Semi-Supervised Training. In Stage-1, we load baseline weights to initialize the student model and teacher model. The offline pseudo labels are generated by the baseline detector and the confidence threshold is set to 0.5 to filter low-confidence false detections. In teacher-student training, the EMA update ratio \(\alpha\) is set to 0.999 and a confidence threshold 0.7 is used for online pseudo labels. We randomly sample 200k unlabeled images for training and the epoch is set to 15. Considering a large number of unlabeled images, we conduct the experiment on 32 NVIDIA V100 GPUs and the batch size is chosen as 64. The pseudo labels switch from offline to online at epoch 5 and the learning rate decays by a factor of 0.1 at epoch 7 and 12.

In Stage-n, we use the semi-supervised model trained in the last stage to update offline pseudo labels on the same unlabeled images and load it to finetune for one more round. Online pseudo labels are not used here and the initial learning rate is set to 0.003.

5.2. Results

We first study the effectiveness of different techniques on training the baseline detector and the results are shown in Table 1. It is easy to see that augmentation strategies to avoid overfitting are necessary for training a good baseline detector as the training set only consists of 5000 labeled images.

In Table 2 we validate the performance of different parts in the O2O semi-supervised framework. For quick experiments, we use the 5k labeled images and randomly sampled 50k unlabeled images for training. Adding online pseudo

| Baseline (Cascade R-CNN) | 67.32 |
| + color jitters | 67.81 (+0.49) |
| + MixUp / CutMix | 68.75 (+0.94) |
| + 50 epochs training | 69.48 (+0.73) |
| + multi-scale training | 70.83 (+1.35) |
| + HTC mask heads | 71.47 (+0.64) |

Table 1. Performance of techniques of a single model with single scale testing on the validation set trained on training set.
For expert models, different resampling methods are used to improve the detection ability on the tricycle category, but none of them are effective enough. We hypothesize that with only a small amount of labeled images, it is hard to train an expert model using only resampling methods to cover the whole feature space of a certain category, as the backgrounds and objects are varied. We also trained another model only for detecting small objects in the image but found out that many more false detections were introduced and the whole performance degraded.

### 6. Conclusion

In this report, we present our method for the semi-supervised object detection task. The proposed O2O framework which combines offline and online methods shows its effectiveness in our experiments. Also, we build a strong baseline detector using augmentation strategies and design a useful auto ensemble scheme. The above works led us to 2nd place in the SSLAD Track 1 Challenge.

### References


