Acta Scientific COMPUTER SCIENCES

Volume 4 Issue 1 January 2022

Research Article

Cascaded Semi-supervised Annotation Tool for Image Data Labeling

Ruixu Liu* and Vijayan K Asari

Department of Electrical and Computer Engineering, University of Dayton, USA

***Corresponding Author:** Ruixu Liu, Department of Electrical and Computer Engineering, University of Dayton, USA.

Received: November 15, 2021
Published: December 13, 2021

Abstract

Object detection algorithms have advanced rapidly during the last decade, especially after realizing the efficiency of Convolutional Neural Networks (CNN) for feature extraction. However, it is a time-consuming and labor-intensive task to annotate the objects' bounding boxes as ground truth data that is necessary to efficiently train the CNN for object detection and recognition tasks. We propose a semi-supervised cascaded bounding box labeling strategy for fast and efficient data annotation for large-scale ground-truth generation. We observed around 80% reduction of workload in data annotation by employing our semi-supervised ground-truth generation method on MS COCO dataset.

Keywords: Object Detection; Bounding Box Labeling; Semi-supervised Method; Data Annotation; Ground-truth Generation

Introduction

Object detection is one of the fundamental tasks of computer vision. Given an input image, an object detector outputs a bounding box wherever an object of interest exists. Object detection has wide applications, including face recognition, pedestrian tracking [1-3], autonomous vehicle navigation [4-7], and surveillance systems [8-10]. Current object detection systems are trained based on supervised learning principles which need human-labeled datasets for training the detection model. The performance of supervised machine learning models relies on the annotated training data. However, the challenge for supervised object detection is to collect large, high-quality labeled datasets with the aim of having a well-performing object detection model. The object learning framework based on deep learning is currently divided into two categories viz., one-stage detectors and two-stage detectors. One-stage detectors directly make a class prediction of objects on each location of the feature maps without the cascaded region classification step such as ones used in the Single Shot multibox Detector (SSD) [11], You Only Look Once unified real-time object detector (YOLO) [12], and RetinaNet [13]. Two-stage detectors first make a region proposal to identify possible Regions of Interest (RoI). Then it extracts features from each proposed region, followed by region classifiers that predict the category of the region, such as the one used in the Faster-RCNN [14]. One-stage detectors are significantly more time-efficient and more applicable to real-time object detection. In contrast, two-stage detectors commonly achieve better detection performance and report state-of-the-art results on public benchmarks.

All of the above object detection algorithms are supervised learning methods that need large amounts of annotated object data for training. For training the object detection architectures such as the Simple Semi-supervised Learning framework (SSL) [15] for object detection and the Un-Biased teacher for semi-supervised object detection (UBteacher) [16], it is necessary to have a large number of training images with ground truth annotations in the form of bounding boxes that are tight rectangles around the objects of interest. In order to train robust detectors, the training data needs to be collected from a large-scale dataset. Traditionally, the annotation problem is solved by brute force approaches like crowd-sourcing by a large group of annotators on a web platform such as the Amazon Mechanical Turk [17]. Examples of some popular large-scale datasets for object detection with labeled data are PASCAL VOC [18], ImageNet [19], and MS COCO [20].

However, crowd-sourcing is also an expensive and time-consuming method. Moreover, for specific datasets like the aerial or medical datasets, crowd-sourcing might not be a feasible approach. For such cases, the researchers need to annotate their data one by one, which is extremely labor-intensive. In order to resolve this crucial issue, the development of a semi-automatic or automatic annotation method drew significant attention among deep learning researchers. In this paper, we propose a Cascaded Semi-supervised bounding box Annotation (CSA) methodology on image datasets for the generation of large amounts of ground truth data. The cascaded annotation approach takes advantage of a trained model to get the pseudo-labels on the unlabeled images and predict the annotation based on a retrained model. When the model is subsequently refined step by step, the need for human intervention for correcting the automatically labeled data is significantly reduced.

In the following sections, we review the related literature on object annotation firstly. Then a detailed discussion of the proposed cascaded semi-supervised annotation method is provided in Section 3. Next, we present our experimental setup with a brief description of the evaluation metrics and processing approaches together with the discussion of experimental results in Section 4. Finally, in Section 5, we present the conclusions and indicate the potential research directions for future work.

Related work

There have been many studies focusing on speeding up the image dataset annotation for the object detection tasks. One of the crowdsourcing methods is crowdsourcing annotations for visual object detection [17]. The three steps involved in this algorithm are drawing, quality verification, and coverage verification. In the drawing step, a worker draws one bounding box around one instance of the given image; in the quality verification step, a second worker verifies whether a bounding box is correctly drawn; and in the coverage verification step, a third worker verifies whether all object instances have bounding boxes.

One of the published methods for semi-automatic annotation is the Faster Bounding Box Annotation method [21] that uses a two-stage semi-automatic approach to speed-up bounding box annotation on labeling small training datasets and correcting network proposals. Our proposed method is related to this concept, but we extend the two-stage approach into a cascaded training loop by incorporating step-by-step training iterations rather than employing only two stages. Another published method is the Iterative Bounding Box Annotation [22] that uses an iterative train-annotation loop, which is intended for efficient image annotation in a small batch of images at a time. This method uses the labeled data without employing any self-learning process.

Our cascaded annotation framework uses an incremental learning approach on a small batch of manually labeled images. Then, it trains a detection model with the labeled data to propose bounding boxes on a batch of unlabeled images. Finally, it requests the annotator to correct possible incorrect bounding boxes or label proposals. Thus, the involvement of human annotators is only in the correction stage.

Methodology

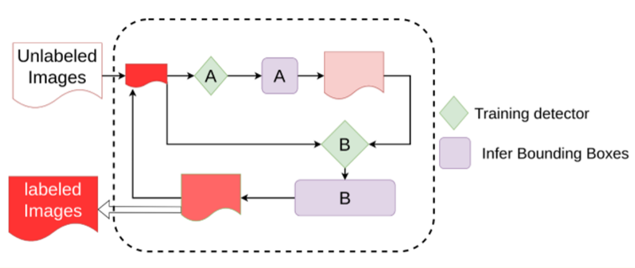
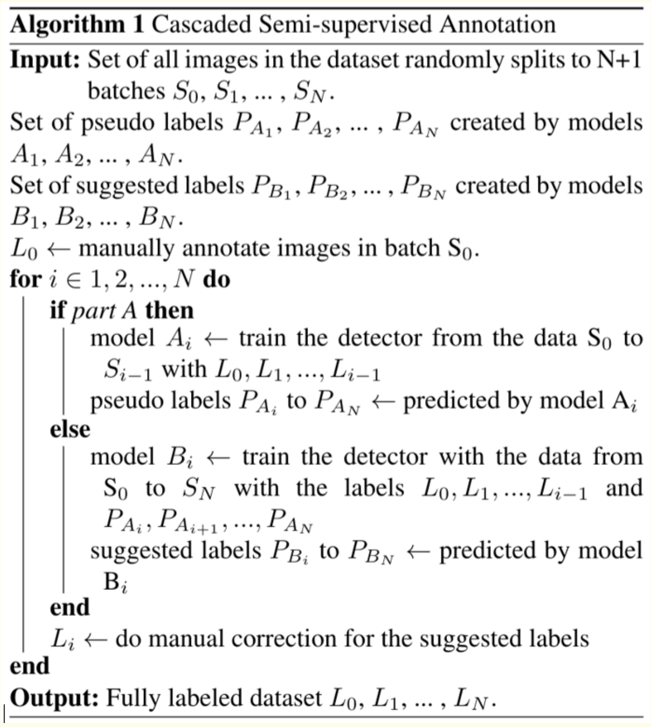


Figure 1: Architecture of the proposed CSA method.

Figure 1 shows the conceptual diagram of our proposed method for cascaded semi-supervised bounding box annotation. The object detection model is trained on a small set of manually annotated images. First, a trained model A is used to predict pseudo labels on all unlabeled data. Next, model B is trained to annotate the unlabeled data combining the labeled and pseudo labeled data. After the first round of train-infer correction, the detector is trained on the recently labeled batch. This process continues in a loop until all unlabeled batches are labeled.

The functional framework of the CSA method uses an incremental learning approach on a small batch of manually labeled images. Then we train a detection model with the labeled data to propose bounding boxes on a batch of unlabeled images and request the annotator to correct possible incorrect bounding boxes or label proposals. In this process, the involvement of human annotators is only in the correction stage. Hence, our method decreases the tedious task of manual annotations. Algorithm 1, shown below, summarizes all the relevant steps of the proposed iterative training method.



```
Algorithm 1 Cascaded Semi-supervised Annotation
Input: Set of all images in the dataset randomly splits to N+1
        batches $S_0, S_1, \dots, S_N$.
Set of pseudo labels $P_{A_1}, P_{A_2}, \dots, P_{A_N}$ created by models
$A_1, A_2, \dots, A_N$.
Set of suggested labels $P_{B_1}, P_{B_2}, \dots, P_{B_N}$ created by models
$B_1, B_2, \dots, B_N$.
$L_0 \leftarrow$ manually annotate images in batch $S_0$.
for $i \in 1, 2, \dots, N$ do
    if part A then
        model $A_i \leftarrow$ train the detector from the data $S_0$ to
          $S_{i-1}$ with $L_0, L_1, \dots, L_{i-1}$
        pseudo labels $P_{A_i}$ to $P_{A_N} \leftarrow$ predicted by model $A_i$
    else
        model $B_i \leftarrow$ train the detector with the data from
          $S_0$ to $S_N$ with the labels $L_0, L_1, \dots, L_{i-1}$ and
          $P_{A_i}, P_{A_{i+1}}, \dots, P_{A_N}$
        suggested labels $P_{B_i}$ to $P_{B_N} \leftarrow$ predicted by model
          $B_i$
    end
    $L_i \leftarrow$ do manual correction for the suggested labels
end
Output: Fully labeled dataset $L_0, L_1, \dots, L_N$.
```

Algorithm 1

The first step in the semi-supervised annotation procedure is to fully annotate (manually)an initial batch of images from the unlabeled dataset. This stage is fully manual and requires human involvement to draw bounding boxes and provide class labels on images. In this stage, we use a basic bounding box annotation tool (LabelImg) with no extra speed-up procedures to create bounding boxes. The second step is to train model A (supervised training) with the fully annotated data (L). Although any detector can be used for this purpose, we focus on the recent deep learning-based object detection models. The third step is to train human-annotated initially labeled data and pseudo labeled data together and relabel the pseudo-labeled data again. Now the system outputs the human-annotated labeled data. Finally, the semi-supervised model suggests labeled data (after predicting the unlabeled data by the network B). Before the cascaded network starts outputting the fully annotated data, the human annotator needs to correct the bounding box labels suggested by model B.

Experimental Results

To evaluate the proposed CSA model on object localization, we must first determine how well the model predicts the object's location. Usually, this is done by drawing a bounding box around the object of interest, but in some cases, it is an N-sided polygon or even pixel by pixel segmentation. The localization task is typically evaluated on the Intersection over Union threshold (IoU) for all of these cases, as shown in figure 2. IOU is obtained by dividing the area of overlap (intersection) between the bounding boxes by the area of the union of the bounding boxes. Considering intersection over union as the most commonly used evaluation metric of object detection, estimating regression quality can judge the IoU between the predicted bounding box and its corresponding ground-truth box. For two bounding boxes, IoU can be calculated as the intersection area divided by the union area. In figure 2, the green box is the predicted box, and the red box is the ground truth box.

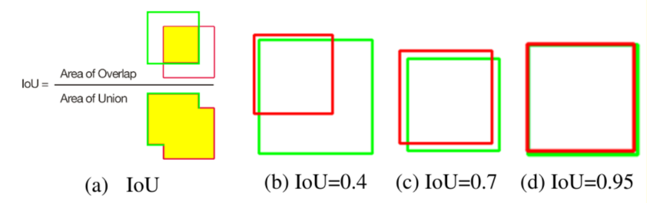


Figure 2: IoU calculation procedure.

Our goal is to estimate the human workload in a simulated setting, attempting to determine how much time a human would spend on a full annotation campaign. We assume the user would correct the annotation if the intersection-over-union (IoU) is less than 0.5. The overlap between the true object location and the predicted bounding box less than 50% is commonly used in object detection performance evaluation. Using IoU, we can identify if the detection (a Positive) is correct (True) or not (False). If IoU is larger than 0.5, it is considered as a true positive (TP), and otherwise, it is a false positive (FP). Given the ground truth (GT) of the dataset, we can compute the true positive (TP), false positive (FP), false negative (FN), and true negative (TN) rates. The precision score is computed as:

$$Precision = \frac{TP}{TP + FP}$$

The precision is to compute the number of true positives relative to the sum of the true positives and false positives. That is the fraction of detected items that are correct. The recall score is computed as:

$$Recall = \frac{TP}{TP + FN}$$

The recall is a fraction of correctly detected items among all the items that should have been detected. The maximum workload (all of the targets are manually labeled) is:

Max workload = # of all targets

The correction workload is the sum of these two steps:

# of corrections =α× # of relabeling +β× # of removals,

where α= 1 and β= 0.5.

The relabeling and removal workloads are computed as:

# of relabeling = FN = # of targets−TP

# of removals = FP = # of detections−TP

The total workload is computed as:

# of workload = #of initial labeling + #of corrections

The human annotation workload for different ratios of training data from the total dataset is shown in figure 3. The remaining data will be used for model prediction to subsequently retrain the model employing human intervention.

It can be seen in figure 3 that as the initial training increases, the workload also increases. Moreover, the more initial data labeled, the better testing performance we can get. The best workload is found for the COCO dataset when the initial training set is between 3% and 10%. When the initial labeling is less than 3%, the workload is also more than the best workload that is observed for 3%.

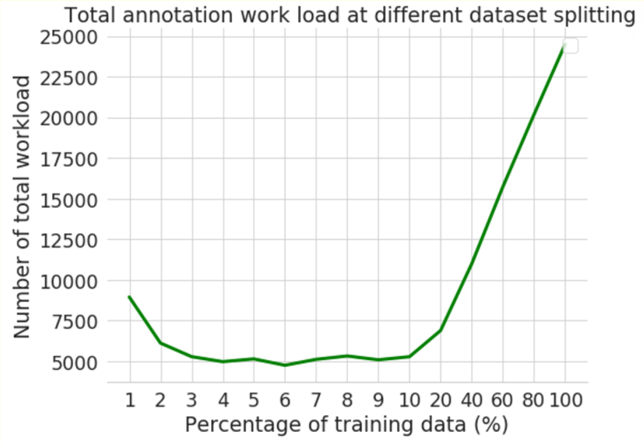


Figure 3: Total annotation workload for different dataset splitting.

Method	Evaluation	1%	5%	10%	20%
Yolo	workload	9051 ± 642	5240 ± 289	5086 ± 141	6986 ± 77
	reduction	63.11% ± 2.6%	78.64% ± 1.2%	79.27% ± 0.6%	71.52% ± 0.3%
Faster-RCNN	workload	7315 ± 314	4795 ± 313	4441 ± 101	5806 ± 95
	reduction	70.18% ± 1.3%	80.45% ± 1.3%	81.90% ± 0.4%	76.33% ± 0.3%

Table 1: Total annotation workload for different methods at different dataset splitting.

We tested our CSA method with two popular detectors, which are Yolo5 and Faster-RCNN with FPN (Feature Pyramid Network) and ROI-align. For each method, we tested 5 times at 1%, 5%, 10%, and 20% initial training data splitting ratios (manually labeled). The quantitative results are shown in table 1. Figure 4 shows that the two-stage method (Faster RCNN) is better than the one-stage method (Yolo5). Furthermore, with the initial labeled data increase, the workload reduction rate is found to be stable.

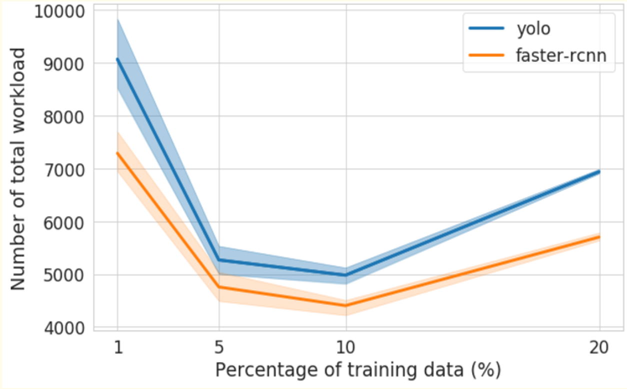


Figure 3: Total annotation workload for different methods at different dataset splitting.

Conclusion

We proposed a similarity-based self-trained approach for the efficient labeling of object detection datasets. The proposed approach can produce high-quality annotation with a reasonable annotation budget. Most of the tedious work is done by the machine, while the human annotator mostly takes care of correction work which is often easier than labeling images from scratch. Extensive experiments on the MS COCO dataset showed that a large amount of manual annotation work could be saved if some focus is paid on sample selection prior to the network training. In the future, we would like to apply this method to more challenging video dataset annotation for semantic segmentation, specifically for the annotation of aerial datasets for lake and pond region segmentation in our ongoing phytoplankton research work. Furthermore, we suggest incorporating unsupervised clustering methods that could reduce human workload even more by automatically selecting the important clusters for initiating the iterative labeling process.

Bibliography

1. Ruixu Liu., *et al*. "Attention mechanism exploits temporal contexts: Real-time 3d human pose reconstruction". Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020).
2. Zhe Cao., *et al*. "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.1 (2019): 172-186.
3. Ruixu Liu., *et al*. "Enhanced 3d human pose estimation from videos by using attention-based neural network with dilated convolutions". *International Journal of Computer Vision* 129.5 (2021): 1596-1615.
4. Ruixu Liu., *et al*. "SLAM for robotic navigation by fusing rgb-d and inertial data in recurrent and convolutional neural networks". 2019 IEEE 5th International Conference on Mechatronics System and Robots (ICMSR). IEEE (2019).
5. Qingquan Li., *et al*. "A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios". *IEEE Transactions on Vehicular Technology* 63.2 (2013): 540-555.
6. Liu Ruixu and Vijayan K Asari. "3D indoor scene reconstruction and change detection for robotic sensing and navigation". Mobile Multimedia/Image Processing, Security, and Applications. International Society for Optics and Photonics 10221 (2017).
7. Liu Ruixu., *et al*. "3D change detection in staggered voxels model for robotic sensing and navigation". Mobile Multimedia/Image Processing, Security, and Applications. International Society for Optics and Photonics 9869 (2016).
8. Aspiras Theus H., *et al*. "Active Recall Networks for Multiperspectivity Learning through Shared Latent Space Optimization". *IJCCI* (2019).
9. Aspiras Theus., *et al*. "Convolutional auto-encoder for vehicle detection in aerial imagery (conference presentation)". Pattern Recognition and Tracking XXX. International Society for Optics and Photonics, 10995 (2019).
10. Liu Ruixu., *et al*. "Deep neural network based approach for robust aerial surveillance". Pattern Recognition and Tracking XXXII. International Society for Optics and Photonics 11735 (2021).
11. Liu Wei., *et al*. "SSD: Single shot multibox detector". European Conference on Computer Vision, Springer, Cham (2016).

12. Redmon Joseph., *et al*. "You only look once: Unified, real-time object detection". Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016).

13. Lin Tsung-Yi., *et al*. "Focal loss for dense object detection". Proceedings of the IEEE International Conference on Computer Vision (2017).

14. Ren Shaoqing., *et al*. "Faster RCNN: Towards real-time object detection with region proposal networks". *Advances in Neural Information Processing Systems* 28 (2015): 91-99.

15. Sohn Kihyuk., *et al*. "A simple semi-supervised learning framework for object detection". arXiv preprint arXiv:2005.04757 (2020).

16. Liu Yen-Cheng., *et al*. "Unbiased teacher for semi-supervised object detection". arXiv preprint arXiv:2102.09480 (2021).

17. Su Hao., *et al*. "Crowdsourcing annotations for visual object detection". Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence (2012).

18. Everingham Mark., *et al*. "The pascal visual object classes challenge: A retrospective". *International Journal of Computer Vision* 111.1 (2015): 98-136.

19. Russakovsky Olga., *et al*. "ImageNet large scale visual recognition challenge". *International Journal of Computer Vision* 115.3 (2015): 211-252.

20. Lin Tsung-Yi., *et al*. "Microsoft COCO: Common objects in context". European Conference on Computer Vision, Springer, Cham (2014).

21. Bishwo Adhikari., *et al*. "Faster Bounding Box Annotation for Object Detection in Indoor Scenes". Proceedings of the 7th European Workshop on Visual Information Processing (EUVIP), July (2018).

22. Bishwo Adhikari and H Huttunen. "Iterative Bounding Box Annotation for Object Detection". Proceedings of the 25th International Conference on Pattern Recognition (ICPR), July (2020).