A Simple and Efficient Method for Bike Parts Detection

Huiming Zhang
Alibaba Group
zhm220845@alibaba-inc.com

Xuan Jin
Alibaba Group
jinxuan.jx@alibaba-inc.com

Pengfei Sun
Alibaba Group
yeqing.spf@alibaba-inc.com

Yuan He
Alibaba Group
heyuan.hy@alibaba-inc.com

Hui Xue
Alibaba Group
hui.xueh@alibaba-inc.com

Abstract

This paper introduces the solution of CS21, the second-place team for VIPriors Object Detection Challenge in 2021. In this work, we build our solution on top of Cascade RCNN (using ResNet50 as the backbone) and with some strategies, including initializing from the self-supervised pre-trained model, soft-NMS with category-related thresholds of IOU, boxes ensemble, and so on. The final result proved the effectiveness of our solution, and the final evaluation score of mAp is 30.3%.

1. Introduction

Deep neural networks have achieved great success in object detection [3, 20, 23, 8, 19, 17, 18] since automatically localizing and detecting an object in images is one of the most important applications of computer vision. And it can be used for visual verification because its task is to locate and identify objects that exist in the image, rather than missing objects. Therefore, visual verification as an automatic visual inspection method, which is widely used in industrial environments, such as infrastructure verification in map-making, missing instrument detection after surgery, part inspections in machine manufacturing, etc.

There are important differences between visual verification and object detection. For object detection, it needs to detect and locate the object in the picture once, while for visual verification, it can perform multiple detections and its focus is whether the object exists.

The VIPriors Object Detection Challenge proposes a novel, specifically created visual object part verification dataset: DelftBikes [12]. And it is used for the detection challenge, whose main objective is to detect bike parts. As shown in the Figure 1, the red boxes are missing parts, and the blue boxes are existing parts, only the existing parts need to be detected. DelftBikes contains 10,000 bike images with 22 densely annotated parts for each bike, dividing into the train, val, and test sets, with 7000,1000 and 2000 images respectively. Besides, they explicitly annotate all part locations and part states as missing, intact, damaged, or occluded. Only the existing parts are evaluated, that is, the intact, damaged, and occluded parts.

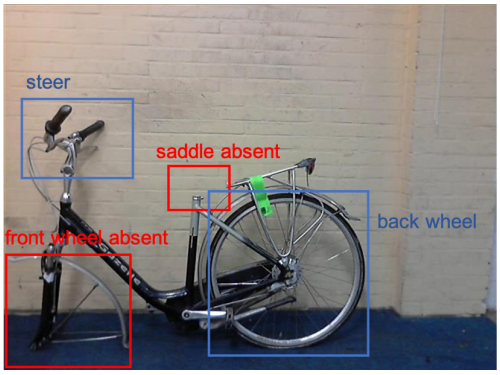

Figure 1. An example image of DelftBikes data set. Bike parts have four states: absent, intact, damaged, or occluded. The missing parts are also bounded with bounding boxes, as shown in the red box. In the testing phase, we only want to detect parts in the other three states, except the absent part.

Existing common object detection datasets such as PASCAL VOC [9], MS-COCO [16], Open Image [13] and Imagenet-det [21] have only the annotation of the whole object, without the annotation of the parts. Pascal-parts [6] and GoCaRD [22] contain part labels, but lack specific location information for the missing object, as is required to evaluate visual verification.

The training set of DelftBike contains 123576 instances, which is relatively small compared with commonly used detection data sets such as MS COCO. In addition, the data set contains a large number of noisy labels, which is mainly due to the inaccurate position of the ground-truth bounding box. As shown in Figure 2, the first line are some example

Figure 2. Example images of DelftBikes visual verification dataset with 22 bounding box annotated parts. A similar posture, orientation, and position can mislead a context-sensitive detector, leading to false detection(the saddle in (b), the wheels in (c,d), etc.). At the same time, there are many wrong bounding box labels, which will also have a larger impact on the convergence of the model(the front_mudguard in (e), the steer in (f), etc.).

images of DelftBikes datasets. The second line are some examples of images with noisy bounding boxes, some of which are in the wrong position, and some bounding boxes contain a lot of background or are just parts of the object.

2. RELATED WORK

Object Detection. There are two main categories of object detection methods: one-stage methods, e.g., [19, 17, 1, 15], and multi-stage methods, e.g., [8, 20, 3]. Multi-stage detectors are usually more flexible and accurate but more complex than one-stage detectors. The output of one-stage object detector can be obtained after only one CNN operation. As for twostage object detector, it usually feeds the high score region proposals obtained from the RPN(region proposals networks) to the secondstage CNN for label prediction and regression.

Self-supervised Learning. The core idea of self-supervised learning is to create free supervisory labels from data, and use the free supervision to obtain generalizable and transferrable representations. Contrastive learning is popular pretext task for self-supervised learning, and recent methods regard it as a dictionary lookup task for training an encoder.

3. METHODS

In this section, we introduce the solution in detail in this competition. It is forbidden to use additional data in this competition, including pre-trained and transformed models. Therefore, we use the data provided by the competition to train a pre-trained model through the self-supervised method. We also applied a series of data augmentation methods to alleviate the problem of insufficient data. In the training stage, we attach DCN [7] module and the Group normalization module [24] to the network model. In the testing phase, soft-NMS is employed instead of original NMS, and different thresholds are set for each category to improve recall rate. Finally, ensemble methods and some effective strategies are applied to bike parts detection.

3.1. Base Detector

The model architecture is base on Cascade RCNN [3] implemented by mmdetection [4]. We use ResNet50 [11] with DCN [7] as our backbone, and used FPN [14] to deal with small instances detection. We use the ResNet50 model instead of a deeper or more complex model because our experiments prove that deeper or more complex networks do not bring gains for object detection. At the same time, we also experimented with other better detection models, such as DetectoRS [18], which worked not well on this data set.

3.2. Self-supervised Pre-training Model

As we all know, it is difficult to converge model training from scratch in the object detection task, especially with insufficient data. The competition prohibits the use of other data than the provided training data, i.e., no pre-training, no transfer learning. However, we can use the data provided by the competition to train a pre-trained model through self-supervised or unsupervised methods, thus making the model converge faster and better. According to the bounding box label information of the trainval set, we cropped and generated 133,245 bike part images for pre-trained model training.

Momentum Contrast(MoCo) [5] is presented for unsu-

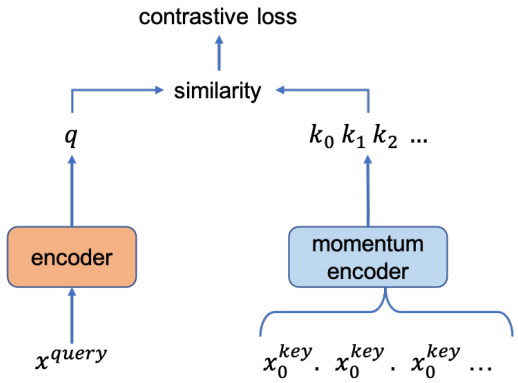

Figure 3. Momentum contrast (MoCo) trains the visual representation encoder by using contrast loss to match the encoded query q with the encoded key dictionary, whose model parameters will be used for initialization of the backbone network of the detection model.

pervised visual representation learning with a contrastive loss [10], which can drive a variety of pretext tasks. MoCo is a mechanism for building dynamic dictionaries for contrastive learning [10], which can be thought of as training an encoder for a dictionary lookup task, as described next. Consider an encoded query q and a set of encoded samples $k_0, k_1, k_2, ...$, positive encoded sample that are similar to q is defined as k_+. As shown in Figure3, contrast learning is to learn an encoder F, which can shorten the distance between q and its positive samplek_+, meanwhile, push the distance between q and its negative samplesk_-. The contrastive loss function is defined as follows:

$$L_q = -log\frac{exp(q \cdot k_+/\tau)}{\sum_{i=0}^{K} exp(q \cdot k_i/\tau)} \tag{1}$$

where τ is a temperature hyper-parameter per [25]. Backpropagation can pass back gradients for all samples by using queues, but the dictionary gets bigger, making updating the encoder tricky. Hence, MoCo proposes a momentum update to address this problem, the parameter update formula is showed in equation 2, query and dictionary encoders are represented as f_q and f_k, respectively. Encoders can be arbitrary convolutional neural networks, such as Resnet and Res2Net.

$$\theta_k = m\theta_k + (1-m)\theta_q \tag{2}$$

where $m \in [0, 1)$ is a momentum coefficient.

MoCo considers a query and a key to be positive if they originate from the same image, or negative sample pairs otherwise. In the experiment, in order to enhance the robustness of the model, different image enhancement methods are randomly selected for the same image to generate positive sample pairs. More importantly, the representations learned by MoCo are well transferable to downstream tasks, as demonstrated in our experiments.

Algorithm 1 Soft-NMS with Category-related Thresholds

Input: $B = \{b_1, ..., b_N\}, S = \{s_1, ..., s_N\}$,
$N_t = \{n_{t1}, ...n_{tc}\}, L = \{l_1, l2, ..., l_N\}$
B is the list of initial detection boxes
S contains corresponding detection scores
N_t is the NMS iou threshold
L contains corresponding detection labels
Output: bounding boxes D and sores S.

```
begin
    D ← {}
    for c in classes do
        B_C = B[L = c], S_C = S[L = c]
        N_c = N_t[c]
        while B_C ≠ empty do
            m ← argmax S_C
            M ← b_m
            D ← D ∪ M
            for b_i in B_C do
                if iou(M, b_i) ⩾ N_c
                    s_i ← s_i f(iou(M, b_i));
        end
    return D,S
end
```

3.3. Soft-NMS with Category-related Thresholds

Non-maximum suppression is an integral part of the anchor-based object detection pipeline. The NMS sorts by score and boxes with high scores suppress boxes with low scores and overlap greater than the threshold. NMS directly deletes the boxes with lower scores, thereby reducing the recall rate of the object detection. There are a lot of ground-truth bounding boxes with noise, so boxes with high scores tend not to fit the target object best. According to the design of the NMS algorithm, if an object is within the predefined overlap threshold, although it is closer to the GT, it will not be recalled.

To this end, we use soft-NMS instead of NMS. Moreover, we found that the coordinate deviations of bounding boxes of each category were different, resulting in that each category was sensitive to the threshold value of IOU. Therefore, we set different IOU thresholds for each category to better improve the recall rate of object detection, as shown in algorithm 1.

3.4. Ensemble Boxes

Ensemble boxes from different object detection models can bring greater performance gains. We trained different neural networks, Double-head Faster-RCNN and Cascade-RCNN, both using resnet50 as the backbone. The exper-

imental results show that the models with different structures have different advantages in detecting different types of objects. As shown in Tabel 1 Double-head Faster-RCNN performs better than Cascade-RCNN for large objects (objects larger than 96 *96), while Cascade-RCNN performs better for small objects (objects smaller than 32 *32) and medium objects. Thus, we fuse the prediction results of different models on the test set to improve performance.

Table 1. **Results of different architecture models**

method	AP(small)	AP(medium)	AP(large)
DH-Faster	13.81	29.09	25.01
Cascade	14.05	29.59	24.23

4. EXPERIMENTS

4.1. Dataset

The DelftBikes [12] dataset contains 22 densely annotated parts, where each part is in one of four possible states: intact, damaged, absent and occluded. All states except absent were used for training and testing. The training set contains 7000 images and 123,576 annotations, the val set contains 1000 images and 17,667 annotations, and the testing set contains 2000 images. In our experiments, only training data were used for model training. The evaluation function was calculated according to mAP(IoU=.50:.05:.95).

4.2. Experiments Setting

We implement our method using mmdetection [4], which is an open-source object detection toolbox. Models are trained using SGD with a momentum of 0.9 and a weight decay of 0.0001. We trained the model for 24 epochs, the learning rate is initialized to 0.08 and decays at a decay rate of 0.1 at 16 epochs and 22 epochs. The ResNet50 backbone is initialized by our self-supervised pretrained model using Momentum Contrast (MoCo) v2 [5].

We adopt a multi-scale augmentation during the training phase. It is emphasized that large-scale training images are very effective for small object detection. Specifically, without changing the aspect ratio, we randomly adjust the short side of the image to 800 1600 pixels, and keep the long side of the image at 2666 pixels. In inference, we adopt multi-scale testing with image sizes, [(2666, 800), (2666, 900), (2666, 1000), (2666, 1100),(2666, 1200), (2666, 1300), (2666, 1400),(2666, 1500), (2666, 1600)], and score threshold of Soft NMS [2] is set to 1e-7. The IOU thresholds of different categories of soft-NMS are individually set.

4.3. Ablation Study

4.3.1 Self-supervised Pre-trained Model

We studied the effect of the self self-supervised pre-trained model. It is prohibited to use of any pre-trained checkpoint, including any pre-trained backbone in this competition. To illustrate the effectiveness of our approach, we compared the pre-trained model generated by ImageNet with our self-supervised pre-trained model. It should be pointed out that the training data of the self-supervised pre-trained model is provided by the competition, without using any additional data. The cropped images of the training data will be used for the training of the self-supervised model. The experimental results shown in Table 2, the performance of the detection model initialized by the self-supervised pre-trained model is 1% higher than that of the unused pre-trained model and 0.3% higher than the ImageNet pre-trained model.

Table 2. **Effect of self-supervised pre-trained model**

method	mAP(val)
no-pretrain	29.64
ImageNet-pretrain	30.14
MOCO v2	**30.79**

4.3.2 Test Time Augmentation

Table 3 shows the impact of the multi-scale(ie. short of the image) in the testing phase. We set the width to 2666 pixels and the height to multiple scales while keeping the aspect ratio of the image unchanged. At the same time, during the test phase, we do not perform any image enhancement, including horizontal and vertical flip. It can be seen from the table that multi-scale testing has a significant improvement in mAP, which increases 0.76 to 30.79 from 30.03. Therefore, our final model used a configuration of 9 scales.

Table 3. **Effect of multi-scale test**

scales	mAP(val)
12k	30.03
11k, 12k, 13k	30.11
10k, 11k, 12k, 13k, 14k	30.72
9k, 10k, 11k, 12k, 13k, 14k, 15k	30.71
8k, 9k, 10k, 11k, 12k, 13k, 14k, 15k, 16k	**30.79**

4.3.3 NMS Methods

We studied the effect of different types of NMS methods. We noticed that mAP improved by 1.0% after using soft-NMS[2]. Since the training data and the test data have noise labels, the ground true bounding box may not fit the target

Table 4. **Results of all useful experiments in val set.**

Method	pre-train	DCN	GN/syncBN	TTA	soft-nms	emsemble	tricks	mAP(val)
Cascade R-50	None	✓	GN					28.40
Cascade R-50	moco_v2	✓	GN					29.14
Cascade R-50	moco_v2	✓	GN		✓			30.03
Double-head R-50	moco_v2	✓	GN	✓	✓			30.41
Cascade R-50	moco_v2	✓	GN	✓	✓			30.79
Cascade R-50	moco_v2	✓	GN	✓	✓	✓		30.91
Cascade R-50	moco_v2	✓	GN	✓	✓	✓	✓	**30.92**

object well, so the soft-NMS can retain the boxes, which have larger iou with GTs. In addition, we found in our experiment that different categories of GT boxes had different deviations from the actual bounding boxes due to the presence of noise labels. Different categories of bounding boxes have different deviations, which are related to the average size of the object. Therefore, we set different IOU thresholds for different categories. What's more, we experimented with different NMS approaches, the results were shown in Table 5. Linear soft-NMS improves by 1% compared to NMS, and our method is further improved by 0.08%.

Table 5. **Effect of different NMS methods**

method	mAP(val)
nms	29.14
soft-nms(gaussian)	30.14
soft-nms(linear)	30.71
soft-nms(Class-related thresholds)	**30.79**

4.3.4 Other Tricks

We ensemble predictions from different models, double-head faster rcnn and cascade rcnn. As shown in Tabel 6, ensemble boxes bring a gain of 0.12%, increasing mAp from 30.79% to 30.91%. After that, we eliminated the overlapping boxes through the spatial position relationship between different categories, which also brought 0.04% improvement.

Table 6. **Effect of other tricks**

boxes emsemble	elimination boxes	mAP(val)
		30.79
	✓	30.83
✓		30.91
✓	✓	**30.92**

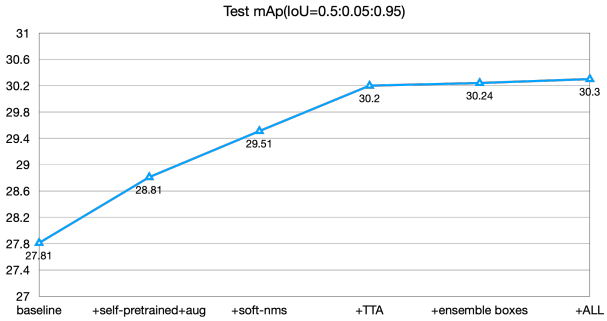

Figure 4. Model performance on the testing set. The +self-supervised pre-train model represents initialize the model from a self-supervised pre-train model. +soft-NMS uses soft-NMS with category-related IOU threshold. +TTA indicates using Test Time Augmentation include multi-scale tests. +ensemble boxes indicates ensembling boxes from various predictions from different models. +All indicates using all the methods and tricks.

4.3.5 Final Results

The final experimental results in val set and test set are shown in Table 4 and Figure 4 respectively. On the basis detector, we added self-supervised pretrain model, GN, soft-NMS, TTA and Emsemble, and finally got 30.92 mAP in Val set and 30.3 mAP in test set.

5. CONCLUSION

In this report, we describe our solution, which was ranked second on Leaderboard on September 24, the normal deadline for the VIPriors object detection challenge. We build our method based on Cascade RCNN, using ResNet50 with DCN as a strong backbone and FPN to cope with small instances. We achieved mAP of 30.3% on the test set, and demonstrated the effectiveness of our approach. Finally, we believe that the organizers will have a fair solution to deal with some unfairness in the competition.

References

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[2] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017.

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

[4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[5] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[6] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1971–1978, 2014.

[7] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.

[8] Xiyang Dai, Yinpeng Chen, Bin Xiao, Dongdong Chen, Mengchen Liu, Lu Yuan, and Lei Zhang. Dynamic head: Unifying object detection heads with attentions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7373–7382, 2021.

[9] Mark Everingham and John Winn. The pascal visual object classes challenge 2012 (voc2012) development kit. *Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep*, 8:5, 2011.

[10] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[11] 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.

[12] Osman Semih Kayhan, Bart Vredebregt, and Jan C. van Gemert. Hallucination in object detection - a study in visual part verification. 2021.

[13] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *International Journal of Computer Vision*, 128(7):1956–1981, 2020.

[14] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[15] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

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

[17] 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.

[18] Siyuan Qiao, Liang-Chieh Chen, and Alan Yuille. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10213–10224, 2021.

[19] 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.

[20] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

[21] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[22] Lukas Stappen, Xinchen Du, Vincent Karas, Stefan Müller, and Björn W Schuller. Go-card–generic, optical car part recognition and detection: Collection, insights, and applications. *arXiv preprint arXiv:2006.08521*, 2020.

[23] Yue Wu, Yinpeng Chen, Lu Yuan, Zicheng Liu, Lijuan Wang, Hongzhi Li, and Yun Fu. Rethinking classification and localization for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10186–10195, 2020.

[24] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[25] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018.