

OOD-CV Challenge Report

October 12, 2022

1 Team details

- Challenge track: Detection Track
- Team name: USTC-IAT-United
- Team leader name: Jun Yu
- Team leader address, phone number, and email: Department of Automation, University of Science and Technology of China, Hefei, Anhui Province, China; +86-13856070316, harryjun@ustc.edu.cn
- Rest of the team members: Keda Lu, Hao Chang, Mohan Jing, Xiaohua Qi, Liwen Zhang, Zhihong Wei, Ye Yu, Fang Gao
- Team website URL: <https://auto.ustc.edu.cn/2021/0510/c25977a484905/page.htm>
- Affiliation: University of Science and Technology of China
- User names on the OOD-CV Codalab competitions: USTC-IAT-United

- Link to the codes of the solution(s):
<https://github.com/sunxin000/ECCV2022-00D-CV-Challenge-detection-Track-USTC-IAT-United>

2 Contribution details

- Title of the contribution: Data-centric approach for OOD detection
- General method description: There are many challenges between the train and test sets of OOD detection dataset, such as unseen distribution and domain shift. Thus we can solve the task which does not have suitable training data to ensure generalization by exploring sample relationships. Among recent data scarcity learning methods, sample relationships have been intensively explored using an explicit scheme from either regularization or knowledge transfer. Specifically, a simple yet very effective way is to directly generate new data samples from existing training data, such as mixup, cutmix, copy-paste, crossgrad.

2.1 Model Design

We consider that the proposals proposed by Region Proposal Network (RPN) are very redundant, especially in the OOD-CV challenge, resulting in poor results by directly using detector with high threshold values. Therefore we use Cascade R-CNN, where cascade regression is used as a resampling mechanism to increase the IoU value of the proposal stage by stage, so that proposals resampled in the previous stage can use the next stage with a higher threshold, and achieve better results.

2.1.1 Backbone

For the choice of backbone, we conducted experimental comparisons with Swin transformer of Local Vision Transformer series, ResNeSt of CNN series, and ConvNeXt, and the experimental results show that

ConvNeXt-Large can get the best results on the OOD-CV Challenge dataset. We guess it may be because ConvNeXt-Large borrows both the design idea of Local Vision Transformer and some tricks of ConvNet, thus has more practical engineering significance.

2.1.2 Balanced Feature Pyramid

We use Feature Pyramid Networks (FPN) for multi-scale feature fusion in order to improve the robustness of the detection algorithm for different size detection targets. We add Balanced Feature Pyramid (BFP) to Feature Pyramid Networks (FPN) to enhance the feature map representation at each level by using multi-level feature map information.

2.1.3 Training Details

When training the model, We use 2x training schedules, and the optimizer took AdamW in order to make the training weights more refined. The optimizer is AdamW while using the cosine annealing learning rate decay method. In order to make the preset boxes match better, We count the size of bbox on the training dataset and reset the size of preset anchor boxes.

2.1.4 SoftNMS

Because of the effect brought by unseen distribution in OOD data, Hard NMS is not suitable for hard selection of prediction frames, so we choose SoftNMS and design detailed ablation experiments.

This continuous function attenuates the detection fraction of non-maximum detection bbox instead of removing them completely. It requires only simple changes to Hard NMS and no additional parameters. In addition, SoftNMS has the same algorithm complexity as Hard NMS and is efficient to use. SoftNMS also requires no additional training and is easy to implement, and it can be easily integrated into the detection process.

2.2 Data Augmentation

2.2.1 Weather Corruption

For changes in the weather dataset, we simulate snow, frost, and fog for data augmentation. After careful study of the validation dataset, we set the severity of fog is 3 or 4, snow is 1 or 2 and frost is 1. The purpose is to simulate the images of a real scene to the maximum extent possible. Figure 1 shows the results of our simulations.

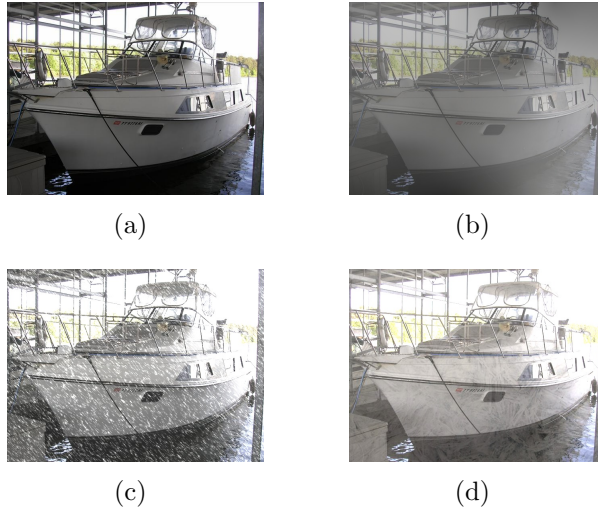


Figure 1: (a) is the image used for training, (b) simulates fog, (c) simulates snow and (d) simulates frost.

2.2.2 Semantic Masking

In particular, to address the problem of occlusion, we also try to occlude the targets on the training images, which effectively improves the detection accuracy. We named it Semantic Masking. Experiments have proven that Semantic Masking is a simple and effective way of data augmentation.

In the following we will explain exactly how it works. We first selected 500 images(in fact, 100 images can also achieve the same effect)

from the ImageNet-1K dataset and manually segmented the objects in the images, which we called semantic masking blocks. After that we performed random resize and rotate, and finally overlaid them on the random positions of the original images. Then the Semantic Masking is finished.

We use semantic masking blocks for occlusion in the following way. First, to set the size of semantic masking blocks, we randomly scale the height of them to 0.8~1.2 times the height of the original image, and the width is scaled according to the height scaling ratio. Then set the rotation of the semantic masking blocks, we will randomly rotate them 20° . Finally, We randomly place them on the center or corner of original images. Figure 2 shows the effect of occlusion.

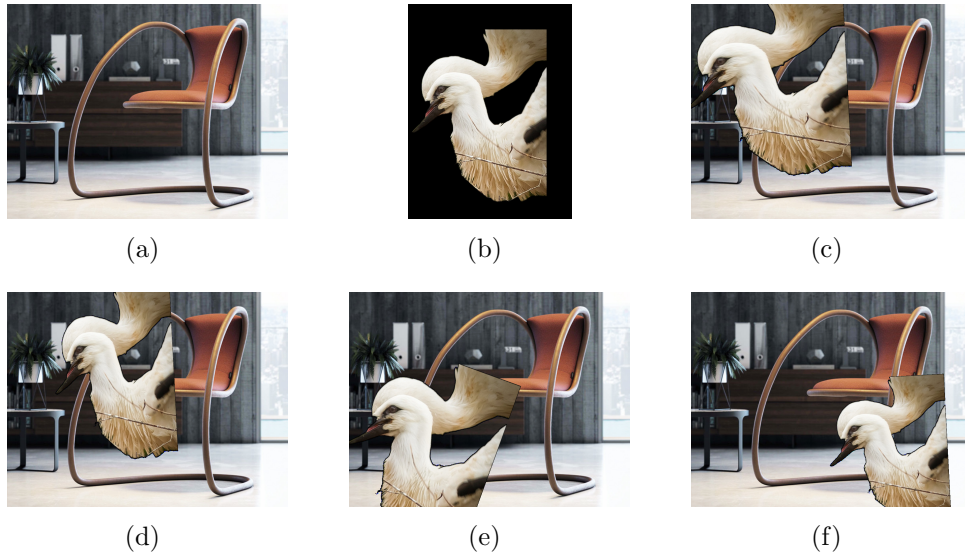


Figure 2: (a) is the image used for training, (b) is the manually segmented semantic masking block. (c~f) are the four images of random occlusion.

2.3 Experiments

We try and evaluate a series of CNN or Transformer based models as feature extraction network, as shown in the Table 1. The advantages of ConvNeXt are obvious, so we choose it as the backbone of the detector.

| Backbone | IID AP50 | OOD AP50 |
|-----------------------|---------------|---------------|
| Convnext-Large | 0.7223 | 0.6399 |
| resnest101 | 0.5743 | 0.6191 |
| Swin-Base | 0.3360 | 0.4158 |

Table 1: The effect of different backbones(phase-1)

After deciding to adopt Cascade RCNN, we tested the IoU threshold of RPN and RCNN. The results are shown in Table 2 and Table 3. When testing the IoU threshold of PRN, the IoU threshold of RCNN is 0.5, while when testing the IoU threshold of RCNN, the RPN IoU is the best experimentally obtained value of 0.7. We also compare the accuracy difference before and after occlusion addition, and the results are shown in Table 4. The experiments show significant improvement on the occlusion dataset after adding occlusion data augmentation.

| RPN IoU threshold | IID AP50 | OOD AP50 |
|-------------------|---------------|---------------|
| 0.5 | 0.6555 | 0.7166 |
| 0.6 | 0.6558 | 0.7161 |
| 0.7 | 0.6563 | 0.7177 |
| 0.8 | 0.6572 | 0.7175 |
| 0.9 | 0.6559 | 0.7175 |

Table 2: The effect of different RPN IoU thresholds(phase-1)

| RCNN IoU threshold | IID AP50 | OOD AP50 |
|--------------------|---------------|---------------|
| 0.3 | 0.6567 | 0.7178 |
| 0.4 | 0.6562 | 0.7180 |
| 0.5 | 0.6563 | 0.7177 |
| 0.6 | 0.6540 | 0.7151 |
| 0.7 | 0.6511 | 0.7096 |

Table 3: The effect of different RCNN IoU threshold thresholds(phase-1)

During testing, we performed extensive experiments of SoftNMS to determine the optimal parameter settings. Table 5 shows our results.

| Backbone | IID AP50 | OOD AP50 | occlusion AP50 |
|---------------------------|---------------|---------------|----------------|
| ConvNeXt | 0.6503 | 0.7116 | 0.4474 |
| ConvNeXt occlusion | 0.6643 | 0.7823 | 0.8069 |

Table 4: The effect of occlusion(phase-1)

| method | min score | IID AP50 | OOD AP50 |
|---------------|-------------|---------------|---------------|
| linear | 0.05 | 0.6546 | 0.7151 |
| linear | 0.01 | 0.6563 | 0.7177 |
| linear | 0.001 | 0.6511 | 0.7067 |
| linear | 0.0001 | 0.6411 | 0.7068 |
| gaussian | 0.05 | 0.6488 | 0.7026 |
| gaussian | 0.01 | 0.6502 | 0.7061 |

Table 5: The effect of different SoftNMS thresholds(phase-1)

| OOD-AP50 | shape | pose | texture | context | weather | occlusion |
|----------|--------|--------|---------|---------|---------|-----------|
| 0.6788 | 0.6509 | 0.7081 | 0.6134 | 0.5926 | 0.6366 | 0.8713 |

Table 6: The final results on the test dataset(phase-2)

- Description of the particularities of the solutions deployed for each of the tracks: In addition to general techniques for model selection and training, in particular we use masking augmentation to improve detection accuracy. Instead of the cutout, cutmix or mixup, which was experimentally proven to be ineffective, we extracted 500 objects from the imagenet-1K named semantic masking blocks and then stretched them proportionally to cover random positions of the training data.
- References:
 1. Cai, Zhaowei, and Nuno Vasconcelos. "Cascade r-cnn: Delving into high quality object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
 2. Dai, Xiyang, et al. "Dynamic head: Unifying object detection heads with attentions." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.
 3. Pang, Jiangmiao, et al. "Libra r-cnn: Towards balanced learning for object detection." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.
 4. Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in neural information processing systems 28 (2015).
 5. Bodla, Navaneeth, et al. "Soft-NMS—improving object detection with one line of code." Proceedings of the IEEE international conference on computer vision. 2017.
 6. Liu, Zhuang, et al. "A convnet for the 2020s." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.
 7. He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
 8. Zhang, Hongyi, et al. "mixup: Beyond empirical risk minimization." arXiv preprint arXiv:1710.09412 (2017).
 9. Yun, Sangdoo, et al. "Cutmix: Regularization strategy to train strong classifiers with localizable features." Proceedings of the

IEEE/CVF international conference on computer vision. 2019.

10. DeVries, Terrance, and Graham W. Taylor. "Improved regularization of convolutional neural networks with cutout." arXiv preprint arXiv:1708.04552 (2017).

11. Loshchilov, Ilya, and Frank Hutter. "Sgdr: Stochastic gradient descent with warm restarts." arXiv preprint arXiv:1608.03983 (2016).

- Representative image / diagram of the method(s): Figure 3 shows our method. In the data processing stage, we perform occlusion of the images as well as fog, frost and snow simulation. When building the Cascade RCNN we replace the backbone with ConvNeXt-Large and add BFP to the FPN layer. In addition we resize the preset anchors according to the analysis of the data. The training schedule is 2x. The optimizer is AdamW and the cosine annealing warm strategy is added. SoftNMS and multiscale tests are also adopted for better detection of targets in the images.

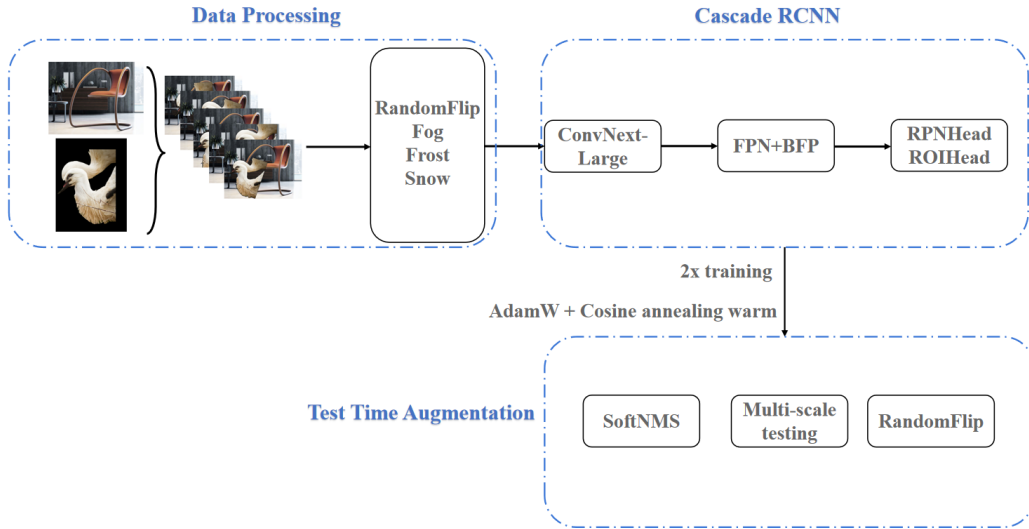


Figure 3: The overall framework diagram of our proposed approach

3 Global Method Description

[* Indicates method used in competition test results.]

- Total method complexity: With 8 GeForce RTX 3090 graphics cards, it takes a total of 15 hours to train 24 epochs and only 20 minutes to infer. The time required can be further reduced by using devices with stronger arithmetic power.
- Model Parameters: Cascade RCNN (ConvNeXt as backbone) param count:241M.
- Run Time: With 8 GeForce RTX 3090 graphics cards, it takes about 40 minutes to train one epoch and at least 15 epochs to train. Inference takes only about 20 minutes.
- Which pre-trained or external methods / models have been used: Pre-trained ConvNeXt-Large using ImageNet-1K dataset.
- Training description: For model selection, we used Cascade RCNN with ConvNeXt as the backbone, and added Balanced Feature Pyramid (BFP) to FPN to enhance the representation of feature maps at each level by using feature map information from multiple levels.
For data augmentation, we first randomly resize the image to [640,1333] or [800,1600]. After that we used imgaug library to simulate the damage of fog, snow and frost. Finally we occlude the images.
For training, we choose AdamW optimizer and use cosine annealing learning rate for training. 2x training strategy is used and the number of training epochs is 24. The best results is obtained at 15 epochs.
- Testing description: We replace SoftNMS with NMS. Through extensive experiments, we determine an IoU threshold of 0.7 in RPN and a threshold of 0.4 in Cascade RCNN. Finally we use a linear function instead of gaussian function for the decay of the confidence in SoftNMS.

- Quantitative and qualitative advantages of the proposed solution: Firstly, ConvNeXt has been proven its effectiveness in this challenge of classification track. Thus we use ConvNeXt as the backbone of Cascade RCNN, which can extract deep semantic information of the images efficiently. Secondly, using semantic masking blocks to mask the original image greatly improves the mAP on occlusion images, while cutout cannot achieve a similar effect. This is mainly because compared to the real occlusion images, the 0 pixels of cutout lack the semantic information that the mask should have.
- Results of the comparison to other approaches (if any): With the exact same parameter settings, we also used DyHead for this task. By applying attention in three different perspectives (scale awareness, spatial location, and multitasking) separately, it unifies the detection head without increasing the computational effort and significantly improves the expression of the detection head. The mAP of DyHead is higher than Cascade RCNN by about 1% in phase-1, but lower in phase-2. In the final test set, Cascade RCNN reached 67.88%, while dyhead only reached 65.08%.
- Novelty of the solution and if it has been previously published: In dealing with occlusion problem, the common data augmentation is generally cutout, cutmix or mixup, which ignore the semantic information of the occluders. Therefore, we extract semantic masking blocks from ImageNet-1K and use them to occlude the original image. Our experiments demonstrate that it effectively enhances the model’s ability to detect the occluded targets.

4 Ensembles and fusion strategies

- Describe in detail the use of ensembles and/or fusion strategies (if any).: The proposals generated by multiple models are fused. The specific fusion method is to fuse all the proposals generated by the models into a model generated proposal. Then all the proposals are processed by SoftNMS or Hard NMS. Finally, the proposals whose

scores are greater than a specific threshold are retained.

- What was the benefit over the single method?: We fusion the output of the Cascade RCNN ConvNeXt -Large and dynamic head ConvNeXt-Large model proposals. Cascade RCNN uses a resampling mechanism of cascade regression to increase the IoU of the proposals stage by stage so that the proposals resampled in the previous stage can adapt to the next with a higher threshold. Dynamic head unifies different target detection heads using the dynamic head framework and attention mechanism. However the score has not been effectively improved after fusion. We analyze that the reason is that the proposal of multiple models is combined, which misleads the processing of the NMS and therefore cannot be effectively fused.
- What were the baseline and the fused methods?: The baseline is Cascade RCNN with ConvNeXt-Large as backbone. The fusion method is to combine the proposals of the Cascade RCNN and Dynamic Head whose backbone is also ConvNeXt-Large, and then retain the proposals whose threshold is greater than 0.2 after the SoftNMS or Hard NMS.

5 Technical details

- Language and implementation details (including platform, memory, parallelization requirements): This method is implemented in python. 8 GeForce RTX 3090 graphics cards which are used for parallel training and testing. Each graphics card occupies approximately 22 GB of video memory for training and 4 GB for testing.
- Human effort required for implementation, training and validation?: We need to perform deep exploratory data analysis at the beginning of the project implementation, but our approach does not require Human effort for training and validation, and the approach can be deployed end-to-end.

- Training/testing time? Runtime at test per image: In the case of 8 GeForce RTX 3090 graphics cards parallel training, it should last for up to 15 hours for 24 epochs. In the case of eight 8 GeForce RTX 3090 graphics cards parallel testing, it takes about 33 minutes to infer in Phase-2, the inferring speed is about 6.64 img/s, and the time required for each image test is 150 ms.
- Comment the efficiency of the proposed solution(s)?: We believe that The solution is still very effective. Firstly, we incorporate ConvNeXt-Large into Cascade RCNN, which increases the complexity compared to ResNet but greatly improves the accuracy of the detection head. Secondly, we don't use pseudo-labeling strategy, which means that excellent results can be obtained by training only once on the training dataset of only eight thousand images.

6 Other details

- General comments and impressions of the OOD-CV challenge.: In real scenes, test dataset and training dataset are often not independently and identically distributed. Therefore, it is of great practical significance to solve this problem, which is conducive to the further implementation of the model and generates more value.
- Other comments: For the exploration of the nature of OOD data, there is no good conclusion at present. I hope our scheme can promote the further development of this field.