Joint COCO and Mapillary Workshop at ICCV 2019: LVIS Challenge Track

Technical Report: Classification Calibration for Long-tail Instance Segmentation

*Tao Wang[1] *Yu Li[2,1] Junnan Li[3] Junhao Liew[1] Sheng Tang[2] Steven Hoi[3] Jiashi Feng[1]

[1]Department of Electrical and Computer Engineering, National University of Singapore, Singapore

[2] Institute of Computing Technology, Chinese Academy of Sciences, China

[3] Salesforce Research Asia, Singapore

twangnh@gmail.com liyu@ict.ac.cn junnan.li@salesforce.com liewjunhao@u.nus.edu

ts@ict.ac.cn shoi@salesforce.com elefjia@nus.edu.sg

Abstract

Remarkable progress has been made in object instance detection and segmentation in recent years. However, existing state-of-the-art methods are mostly evaluated with fairly balanced and class-limited benchmarks, such as Microsoft COCO dataset [7]. In this report, we investigate the performance drop phenomenon of state-of-the-art two-stage instance segmentation models when processing extreme long-tail training data based on the LVIS [5] dataset, and find a major cause is the inaccurate classification of object proposals. Based on this observation, we propose to calibrate the prediction of classification head to improve recognition performance for the tail classes. Without much additional cost and modification of the detection model architecture, our calibration method improves the performance of the baseline by a large margin on the tail classes. Codes will be available.[1].

1. Experimental Details

Dataset statistics Different from [5], we divide all the 1,230 categories of the LVIS v0.5 dataset into 4 sets, which respectively contain <10, 10-100, 100-1,000 and $>1{,}000$ training object instances. We denote them as subset (0, 10), subset [10, 100), subset [100, 1000) and subset [1000, -] for convenience of expression. Please see Table 1 for detailed statistics. Beyond the test set results, we evaluate model performance based on such category split in this report, in order to see the effect of training instance number and analyze the long-tail object instance detection models. We claim that the improvement on the tail bin, i.e. subset (0, 10), of the validation set does not contribute much to the overall AP as it contains only 67 classes, though the category distribution of the test set is unknown.

Sets	$(0, 10)$	$[10, 100)$	$[100, 1000)$	$[1000, -]$	total
Train	294	453	302	181	1230
Train-on-val	67	298	284	181	830

Table 1: Category division based on training instance number. *Train-on-val* means the subset of categories that appear in the validation set.

Training and Evaluation Our implementation is based on the mmdetection toolkit [4]. Unless otherwise stated, the models are trained on LVIS-v0.5 training set and evaluated on LVIS-v0.5 validation set for mask prediction tasks. The external data used in the experiments are introduced in Sec. 4. All the models are trained with SGD, 0.9 momentum and 8 images per minibatch. The training schedule is 8th/11th/12th epoch updates with learning rates of 0.01/0.001/0.0001 respectively, unless otherwise stated.

2. Classification Calibration

We first investigate the performance degradation of the baseline Mask-RCNN [6] on tail classes. Then, based on our observations for the possible causes of this phenomenon, we propose a classification calibration method for improving the model performance over tail classes.

2.1. Missed Detection of Tail Classes

For simplicity of analysis, we train a baseline Mask R-CNN with ResNet50-FPN backbone and *class agnostic box and mask heads*. As shown in Table 2, the model performs

[1]Both * authors contributed equally to this work. Our team name on EvalAI is **lvlvisis**

Model	$AP_{(0,10)}$	$AP_{[10,100)}$	$AP_{[100,1000)}$	$AP_{[1000,-]}$	AP
mrcnn-r50-thr	0.0	5.4	16.6	25.1	13.1
mrcnn-r50	0.0	13.3	21.4	27.0	18.0

Table 2: Performance of baseline Mask-RCNN with *class agnostic box and mask heads* on validation set. mrcnn-r50-thr means testing with 0.05 detection threshold and mrcnn-r50 denotes testing with 0.0 threshold.

Dataset	AP	AR_{1k}
LVIS	18.0	51.0
COCO	32.8	55.9

Table 3: Comparison of baseline models trained on COCO and LVIS. Models are all evaluated with the 5k validation set. AR_{1k} denotes average recall at 1000 proposals. COCO results are measured on minival set.

Model	$AP_{(0,10)}$	$AP_{[10,100)}$	$AP_{[100,1000)}$	$AP_{[1000,-]}$	AP
mrcnn-r50	0.0	13.3	21.4	27.0	18.0
props-gt	39.7	45.1	31.4	29.3	36.6

Table 4: Test with ground truth labels of proposals.

poorly, especially on the tail sets (0,10] and (10, 100]. Even when we lower the detection threshold to 0, which improves 5% mAP, the mAP for the subset (0, 10] is still 0. This result reveals that the Mask-RCNN model trained with normal setting is heavily biased toward the many-shot classes (i.e. those with more training instances).

We then calculate the proposal recall of the model and compare it with that of the same model trained on COCO dataset. As shown in Table 3, the same baseline model trained on LVIS suffers a drop of 8.8% in the proposal recall compared with that on COCO, and notably, a 45.1% drop in the overall AP. This indicates the degradation of proposal classification accuracy is the major cause of final performance drop on long-tail training data.

To verify our observation, for RPN generated proposals, we assign their ground truth class labels and evaluate the AP, instead of using the predicted labels. As shown in Table 4, AP on tail classes is increased by a large margin, especially on the (0,10) and [10, 100) bins. This confirms the observation that the low performance over tail classes is mainly caused by the inability of the model to recognize their correct categories from current generated proposal candidates.

2.2. Classification Calibration with Retrained Head

To improve the performance of the second stage over tail classes, our strategy is to retrain the classification head with data obtained by class balanced sampling and combine predictions of the new classification head with the original one. This approach, though simple, can effectively improve the recognition accuracy on tail classes while maintaining good performance on many-shot classes. We name it *classification calibration*.

Concretely, we sample a fixed number of classes for each step, and sample one image corresponding to each of the sampled classes. In our current implementation, 16 classes and 1 image per class are sampled. The sampled images are fed to the trained model, and the obtained proposals are matched with ground truth boxes using the same IOU threshold as the original detection model training. Only the proposals corresponding to the sampled classes are selected, together with the ground truth boxes of these classes, for training the new head; the other proposals are ignored. During training, we keep the parameters in the backbone network and RPN frozen.

As shown in Table 6, with the newly trained head as the proposal classifier, AP on tail-class bins (0, 10) and [10, 100) is boosted by a large margin. However, due to insufficient training on many-shot classes, AP on [100, 1000) and [1000,-] drops significantly. To enjoy the advantages of both new and original heads, we have tried many different ways to combine their predictions. Refer to Table 6 for details. We find that simply concatenating the predictions of the new head on tail classes ((0, 10) and [10, 100)) with those of the original head on many-shot classes ([100, 1000) and [1000,-]) yields the best results overall.

2.3. Generalization to Multi-stage Cascaded Models and Large Backbones

To further improve the overall performance, we apply the proposed calibration method to multi-stage cascaded models with more complex architectures. State-of-the-art cascaded model Hybrid Task Cascade [3] (HTC) is utilized here. We find that HTC brings a large improvement over vanilla Cascaded Mask-RCNN [2] on LVIS dataset. See Table 5 for details.

All the three classification heads in the three stages of the HTC framework are retrained with our proposed sampling strategy, and we average the predictions of these three new heads during inference following the original setting. Then, the predictions on tail classes are concatenated with the original classification results. Table 7 shows the results of our calibration method applied to HTC with ResNeXt-101-64d backbone. The scores of categories for (0, 10) bin which are predicted by the new head are concatenated with the scores of other categories predicted by the original head.

We think it is more reasonable to take into consideration

Models	COCO		LVIS	
	box	mask	box	mask
cascaded-mrcnn	45.4	39.1	28.6	25.9
htc	46.9	40.8	31.3	29.3

Table 5: Comparison of cascaded Mask-RCNN and Hybrid Task Cascade (HTC) on COCO and LVIS dataset validation set. The two models use the same backbone ResNeXt-101-64x4d. They are trained with 20 epochs and learning rate decay at 16th and 19th epoch.

Model	$AP_{(0,10)}$	$AP_{[10,100)}$	$AP_{[100,1000)}$	$AP_{[1000,-]}$	AP
mrcnn-r50	0.0	13.3	21.4	27.0	18.0
rhead-only	8.5	20.8	17.6	19.3	18.4
rhead-avg	8.5	20.9	19.6	24.6	20.3
rhead-det	8.6	22.0	16.7	25.2	19.8
rhead-cat	**8.6**	**22.0**	**19.6**	**26.6**	**21.1**
rhead-cat-thr	8.5	20.8	20.1	26.7	20.9
rhead-cat-scale	8.5	21.3	19.9	26.7	21.0

Table 6: Different ways for calibrating predictions of original classification head with newly trained head. The ways we have tried include rhead-only (using only newly trained head predictions), rhead-avg (averaging predictions of original head and new head), rhead-det (using the two heads separately for detection outputs and combining them afterward, i.e., two expert models), rhead-cat (simply concatenating tail classes predictions of new head and many-shot classes predictions of original head, with (0,10) and [10, 100) for new head and [100, 1000) [1000,-] for original head), rhead-cat-thr (filtering new head predictions with 0.05 threshold and then concatenating), and rhead-cat-scale (scaling new head predictions by ratio of average background score between new and original head predictions).

the number of classes in each bin in long-tail detection evaluation, rather than just averaging AP of all classes. This is because, the number of classes in each bin may vary largely and the bins with fewer classes tend to be down-weighted in overall mAP. In this sense, the importance of solving the tail problem is not obviously and directly demonstrated by using the current evaluation metric mAP. For example, the validation set of LVIS v0.5 contains only 67 classes with less than 10 training instances, while the numbers are much larger for the [10, 100) bin and [100, 1000) bin, which are 298 and 284 respectively. The improvement on (0, 10) bin would be down-weighted in mAP.

Model	$AP_{(0,10)}$	$AP_{[10,100)}$	$AP_{[100,1000)}$	$AP_{[1000,-]}$	AP
htc-x101	7.1	30.5	30.7	33.9	29.4
calibration	**16.0**	30.6	29.8	33.5	29.8
htc-x101-ms-dcn	5.6	33.0	33.7	37.0	31.9
calibration	**12.7**	32.1	33.6	37.0	32.1

Table 7: Results of applying our calibration to state-of-the-art multi-stage cascaded instance segmentation model Hybrid Task Cascade (HTC).

Model	$AP_{(0,10)}$	$AP_{[10,100)}$	$AP_{[100,1000)}$	$AP_{[1000,-]}$	AP
mrcnn-r50	0.0	13.3	21.4	27.0	18.0
img-sample	7.7	23.2	21.4	26.2	22.0
calibration	8.6	22.0	20.2	26.7	21.3
htc-x101	5.6	33.0	33.7	37.0	31.9
img-sample	10.3	32.4	33.4	36.6	31.9
calibration	12.7	32.1	33.6	37.0	32.1

Table 8: Comparison with image level sampling trained model on baseline Mask R-CNN.

2.4. Comparison with Image-level Repeat Sampling

As shown in Table 8, we compare our classification calibration approach with image-level repeat sampling for the whole network, which is reported as the best baseline in [5]. Although our calibration method has lower overall mAP on validation set than image-level repeat sampling on tail classes, it has higher performance on the most tail bin (0, 10). When generalized to the more complex multi-stage model HTC, our method performs better. The performance of our calibrated model suffers less drop on many-shot classes and enjoys much improvement on tail-classes than image-level repeat sampling method.

3. Final Models and Test Set Submission

As shown in Table 9, our final submitted results on the test set are from the ensemble of 4 models with different backbones. However, due to time limit, we only have our best single model (31.9 AP on val) calibrated among all final models. We believe the final ensemble results will be stronger on tail classes if all models are calibrated.

4. External Data

Microsoft COCO dataset [7] (2017 version) and COCO-stuff dataset [1] are used as external data for our submitted results. All COCO, COCO-stuff, and LVIS datasets share the same training images but own different annotations (LVIS only uses part of the training images in COCO train2017). We only use the training set of COCO and

Model	val-set
htc_x101_64d_ms_dcn	31.9
htc_x101_32d_ms_dcn	31.4
htc_x101_64d_ms_dcn_cos	30.7
htc_r101_ms_dcn	30.0
ensemble-with-calibration	34.2
add-multiscale-testing	35.2

Table 9: Final models performance and ensemble results on validation set. ms denotes multi-scale training, dcn means deformable convolution and cos means cosine learning rate schedule.

Model	AP	AP_r	AP_c	AP_f
best-baseline	20.5	9.8	21.1	30.0
w/o	22.89	5.90	25.65	35.26
with-calibration	23.94	10.31	25.26	35.16
ensemble-with-calibration	26.11	11.94	27.98	37.05
add-multiscale-testing	26.67	10.59	28.70	39.21

Table 10: Comparison of baseline model without and with our proposed calibration method on LVIS *test set*. Best-baseline denotes best baseline performance reported [5]; w/o denotes our best single model (31.9 AP on validation set); with-calibration means adds calibration to the model; ensemble-with-calibration means using the ensemble of all models and adding calibration; add-multiscale-testing denotes adding multi-scale testing.

Model	P	S	$AP_{(0,10)}$	$AP_{[10,100)}$	$AP_{[100,1000)}$	$AP_{[1000,-]}$	AP
HTC-x50-fpn			1.4	23.9	25.3	30.4	24.0
HTC-x50-fpn	✓		3.9	25.0	27.0	31.2	25.3
HTC-x50-fpn	✓	✓	3.7	27.2	27.4	31.8	26.4

Table 11: Effect of external data. *P* stands for using COCO box and polygons for pre-training, and *S* for using COCO-stuff pixel-level semantic segmentation label for semantic head of HTC.

COCO-stuff, which contains 118K images. COCO covers 80 thing classes, the same as COCO-stuff, but the latter also contains 91 stuff classes. For COCO, instance-level boxes and polygons are used to pre-train HTC models. We initialize our model with a model pre-trained on COCO. For COCO-stuff, pixel-level semantic segmentation labels are used for training the semantic head of HTC. Table 11 shows the results of using COCO pre-training and semantic head.

5. Conclusion

We propose a classification calibration method for improving the performance of current state-of-the-art proposal based object instance detection and segmentation models over long-tail distribution data. It is able to effectively improve the classification performance over the long-tail distribution data by enhancing the proposal classification accuracy. Currently, our retraining strategy for proposal classification head is not optimized, which we will investigate in future works. For example, we may combine our method with image-level sampling, choose new head designs or use new head training sampling methods, trying to further boost the model performance on long-tail data distribution.

References

[1] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. COCO-Stuff: Thing and stuff classes in context. In *CVPR*, 2018. 3

[2] 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. 2

[3] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4974–4983, 2019. 2

[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. 1

[5] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 1, 3, 4

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[7] 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 *ECCV*, 2014. 1, 3