The Devil is in Classification: A Simple Framework for Long-tail Instance Segmentation — Supplementary Material

Tao Wang[1,4][0000−0002−2480−878X], Yu Li[2,4], Bingyi Kang[4], Junnan Li[3], Junhao Liew[4], Sheng Tang[2], Steven Hoi[3], and Jiashi Feng[4]

[1] NGS, National University of Singapore, Singapore twangnh@gmail.com
[2] Institute of Computing Technology, Chinese Academy of Sciences, China {liyu,ts}@ict.ac.cn
[3] Salesforce Research Asia, Singapore {junnan.li,shoi}@salesforce.com
[4] ECE Department, National University of Singapore, Singapore {kang,liewjunhao}@u.nus.edu elefjia@nus.edu.sg

A Implementation Details

We use PyTorch to implement the proposed method. Our implementation is based on the mmdetection [1]. All the models are trained with SGD and momentum of 0.9, on 8 NVIDIA Tesla V100 GPUs.

A.1 Standard Model Training

For standard model training, i.e., normal training of whole model with random sampling as shown in Fig.2 (a), 8 images per mini-batch are sampled.

For the Mask-RCNN standard model training, the learning rate starts with 0.01 and decays at the 8th and 11th epochs by a factor of 0.1. The training ends at the 12th epoch. The short edge of images is fixed at 800 pixels and the long edge is capped at 1,333 pixels, without changing the aspect ratio.

For the HTC standard model training, the learning rate starts with 0.01 and decays at the 16th and 19th epochs by a factor of 0.1. The training ends at the 20th epoch. We also add multi-scale augmentation for HTC model training. More specifically, the size of image short edge is randomly sampled from [400, 1,400], and the long edge is capped at 1,600 pixels, without changing the aspect ratio.

A.2 Calibration Training

For calibration training, we sample 16 classes and 1 image per class in one mini-batch. proposals are matched to ground truth bounding boxes with threshold of 0.5, as in [3]. We sample the same number of background ROIs as the the foreground ROIs (i.e., # background : # foreground = 1:1). For both Mask-RCNN and HTC calibration, the learning rate starts with 0.01 and decays at the 8000th (i.e., 0.001) and 11000th (i.e., 0.0001) steps by a factor of 0.1. The calibration training ends at 12000 steps.

A.3 Other Hyperparameters

All the hyperparameters for the adopted long-tail classification methods are tuned by grid search. (i) For re-weighting method, N (i.e., the numerator of class dependent loss weight, as in Sec. 4.1) and background loss weight are set as 100 and 1 respectively. (ii) For Focal loss, γ and α are set as 3 and 0.5 respectively. Different from the observation made in [2] that one-stage detector's performance is relatively robust to the value of γ in a wide range, we find the performance of Mask R-CNN model on LVIS is sensitive to the value of γ and $\gamma = 3$ gives the best performance. The hyperparameter C for class-aware margin loss is set as 6.0.

B Qualitative Results

Some qualitative results can be found in Fig. 1.

Fig. 1: Qualitative results for low-shot categories (in [1, 10) bin) on LVIS with r50-ag model. Only relevant detections of low-shot classes are visualized for a better view. Although there are some false positives, the model after calibration can detect and segment those object instances. For the original model without calibration, all those objects are missed. Note some detections have 0.00 score as we only round the score to 2 decimal places

C COCO-LT Sampling

Here we explain how we created the COCO-LT dataset. We evenly divide the 80 categories into 4 subsets according to their category index, i.e. (1-20, 21-40, 41-60, 61-80) respectively, each with 20 classes. For the ith subset if $i \neq 1$, we randomly sample n_i instances with $n_i \in (8 * 10^{4-i}, 8 * 10^{5-i})$. For the first subset (with category indices of 1-20, i.e., i=1) we do not perform sampling. If an instance is not sampled, we remove it from the annotation of its image. For training images without sampled instances, we remove these images. The

category distribution after sampling (COCO-LT as shown as in Fig. 3) follows a long-tail distribution. The validation set is kept as the original and used for evaluation.

D How to Combine the Dual Heads

In addition to the proposed simple threshold selection scheme for combining the calibrated and original heads' prediction, we also explored some other possible schemes. As shown in Table 1, we compare the proposed combination scheme with other alternatives, including (i) *cal-only*: using only the prediction from the calibrated head; (ii) *avg*: averaging predictions of the original head and calibrated head, this is widely adopted way of ensembling two classification models; (iii) *det*: using the two heads separately for detection outputs and combining them afterward (i.e., with NMS), this is most simple but effective way of ensembling detection models; (iv) *sel*: the proposed output combining scheme; (v) *sel-thr*: filtering the calibrated head predictions with 0.05 threshold before *sel*, aiming to reduce low quality detections from calibrated head with low confidence score; (vi) *sel-scale*: scaling calibrated head's predictions by ratio of average background score between calibrated and original head's predictions before *sel*, since the calibrated head is trained with different background and foreground sample ratio, it has different average foreground score compared to original head, *sel-scale* aims to scale alleviate the difference; (vii) *sel-norm*: normalizing the prediction by the summed score over classes after *sel*, aiming to convert the prediction to a normalized classification score.

The proposed combining scheme achieves the best overall result (i.e., 21.1 AP) compared with the other alternatives. Those prior strategies of *sel-thr*, *sel-scale* and *sel-norm* all have similar but slightly lower performance. The result verifies the simplicity and effectiveness of proposed combining method.

Table 1: Ablation result for different ways of combining calibrated and original heads' predictions. The model is Mask R-CNN with ResNet50-FPN backbone and class agnostic box and mask heads. The experiment is with 2 layer fully connected head with random initialization (i.e., 2fc)

Model	AP_1	AP_2	AP_3	AP_4	AP
orig	0.0	13.3	**21.4**	**27.0**	18.0
cal-only	8.5	20.8	17.6	19.3	18.4
avg	8.5	20.9	19.6	24.6	20.3
det	8.6	22.0	16.7	25.2	19.8
sel	**8.6**	**22.0**	19.6	26.6	**21.1**
sel-thr	8.5	20.8	20.1	26.7	20.9
sel-scale	8.5	21.3	19.9	26.7	21.0
sel-norm	8.5	21.9	19.5	26.2	20.9

E How Calibration Learning Rate Affects Performance

While we use the same initial learning rate for calibration as standard model training (i.e., starting from 0.01), we examine the results when varying the initial learning rate for calibration, to see if optimal learning rate for calibration is different from standard model training. We measure model performance with overall AP. As shown in Table 2, we tested the following learning rates of 0.001, 0.002, 0.004, 0.008, 0.01, 0.02, 0.04 and 0.08. The decaying step and factor remains the same. The best performance is achieved at learning rate of 0.01, same as standard model training. The phenomenon also stands in contrast to the observation in conventional fine-tuning that a much lower learning rate compared to pre-training is required for optimal performance.

Table 2: Ablation study for calibration learning rate. The model is Mask R-CNN with ResNet50-FPN backbone and class agnostic box and mask heads

lr	0.001	0.002	0.004	0.008	0.01	0.02	0.04	0.08	baseline
AP	19.5	21.8	21.9	22.0	**22.2**	21.5	21.1	21.0	18.0

F Layers to Perform Calibration

While we calibrate the whole 3-layer fully connected classification head, we tried to only perform the calibration on last layer, and last 2 layers, to see if we can achieve better performance. We measure model performance with overall AP. Table 3 shows the result comparison of those settings. Best result is obtained when we calibrate the whole classification head.

Table 3: Ablation study for calibration layers. lastfc means only calibrate the last fc layer of classification layerl; last2fc indicates calibrating the last 2fc layers; all 3fc means normal setting of calibrating the full 3 layer classification head. The model is Mask R-CNN with ResNet50-FPN backbone and class agnostic box and mask heads

layer	$lastfc$	$last2fc$	$all3fc$	baseline
AP	21.9	21.8	**22.2**	18.0

G LVIS Mean and Std Analysis

As shown in Tab 4, we report mean and std analysis for the major two *SimCal* models on LVIS dataset. The result is suggested by one reviewer in rebuttle period, due to space limit, we place it in supplementary file. As shown from the results, the variance of AP result is much larger for the tail classes while smaller for many-shot classes as they have ample training instances.

Table 4: Mean and standard deviation analysis for *SimCal* models on LVIS dataset. The result is obtained by repeat each experiments 5 times and calculating the mean and standard deviation of each metric

Model	AP_1	AP_2	AP_3	AP_4	AP_r	AP_c	AP_f	AP
r50-ag-lvis	$13.0_{\pm 1.5}$	$23.2_{\pm 0.8}$	$20.7_{\pm 0.3}$	$26.2_{\pm 0.2}$	$18.0_{\pm 0.9}$	$21.3_{\pm 0.3}$	$24.8_{\pm 0.1}$	$22.4_{\pm 0.3}$
r50-lvis	$10.2_{\pm 1.3}$	$23.9_{\pm 0.6}$	$22.5_{\pm 0.4}$	$28.7_{\pm 0.1}$	$16.4_{\pm 0.7}$	$22.5_{\pm 0.4}$	$27.2_{\pm 0.2}$	$23.4_{\pm 0.2}$

References

1. Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C.C., Lin, D.: MMDetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155 (2019)
2. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. IEEE transactions on pattern analysis and machine intelligence (2018)
3. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)