Joint COCO and LVIS workshop at ECCV 2020: LVIS Challenge Track Technical Report: Seesaw Loss for Long-Tailed Instance Segmentation

Jiaqi Wang[1], Wenwei Zhang[2], Yuhang Zang[2], Yuhang Cao[1],
Jiangmiao Pang[3], Tao Gong[4], Kai Chen[1],
Ziwei Liu[1], Chen Change Loy[2], and Dahua Lin[1]

[1] The Chinese University of Hong Kong [2] Nanyang Technological University
[3] Zhejiang University [4] University of Science and Technology of China
Team: MMDet, Account: MMDet, E-mail: wj017@ie.cuhk.edu.hk

Abstract. This report presents the approach used in the submission of the LVIS Challenge 2020 of team **MMDet**. In the submission, we propose **Seesaw Loss** that dynamically rebalances the penalty to each category according to a relative ratio of cumulative training instances between different categories. Furthermore, we propose **HTC-Lite**, a light-weight version of Hybrid Task Cascade (HTC) which replaces the semantic segmentation branch by a global context encoder. Seesaw Loss improves the strong baseline by **6.9% AP** on LVIS v1 *val* split. **With a single model**, and **without using external data and annotations** except for standard ImageNet-1k classification dataset for backbone pre-training, our submission achieves **38.92% AP** on the *test-dev* split of the LVIS v1 benchmark.

1 Methodology

1.1 Seesaw Loss

Existing object detectors struggle on long-tailed datasets, exhibiting unsatisfactory performance on rare classes. We observe that the detector's classifier tends to predict higher confidence for frequent classes and lower scores for rare classes. Note that a training sample for positive class is also a negative sample for other classes in a multi-class classifier. The overwhelming number of samples in frequent classes leads to models whose rare class confidences are severely suppressed.

To tackle this problem, we propose **Seesaw Loss** for long-tailed instance segmentation. Seesaw Loss dynamically re-balances the penalty to each category during training, according to a relative ratio of cumulative training instances between different categories. Seesaw Loss has three properties. 1) Seesaw Loss is dynamic *w.r.t.* the relative ratio between categories. It dynamically modifies the penalty according to the relative ratio of instance numbers between each category pair rather than split categories into different groups [12,19]. 2) Seesaw Loss is smooth and makes no clear distinction between frequent and rare classes. It smoothly adjusts the punishment on rare classes when the training instances are positive samples of other relatively frequent classes. 3) Seesaw

Loss is self-calibrated so that it can be applied in a distribution-agnostic manner. It directly learns to balance the penalty to each categories during training, without relying on known dataset distributions [1,5,15,19] or a specific data sampler [7,9].

Seesaw Loss. Seesaw Loss can be derived from cross-entropy loss whose general formulation can be written as

$$L_{cls}(z) = -\sum_{i=1}^{C} y_i \log\left(\sigma_i\right), \text{ with } \sigma_i = \frac{e^{z_i}}{\sum_{j=1}^{C}\left(1-y_j\right) S_{ij} e^{z_j} + e^{z_i}}, \tag{1}$$

where $z = \mathcal{W}^T x + b$ is the activation of classifier, $S_{ij} = 1$ for cross-entropy loss and $y_i, i \in C$ is the label.

Seesaw loss accumulates the number of training samples for each category $N_i, i \in C$ during each training iteration. Given an instance with positive label i, for the other category j, Seesaw Loss dynamically adjusts the penalty for negative label j *w.r.t.* the relative ratio of accumulated training samples $\frac{N_j}{N_i}$ as

$$S_{ij} = \begin{cases} 1, & \text{if } N_i \leq N_j \\ \left(\frac{N_j}{N_i}\right)^p, & \text{if } N_i > N_j \end{cases} \tag{2}$$

When category i is more frequent than category j, Seesaw Loss will reduce the penalty on category j for samples of category i by a factor of $\left(\frac{N_j}{N_i}\right)^p$, like a seesaw. The exponent p adjusts the scale and is set to 0.8 in experiments. If category i is far more frequent than category j, the punishment will be significantly alleviated to protect the category j. Otherwise, Seesaw Loss will keep the penalty on negative classes to reduce misclassification.

Classifier Design. Different from traditional detectors which predict classification activation as $z = \mathcal{W}^T x + b$, we adopt a normalized linear layer as

$$z = \tau \widetilde{\mathcal{W}}^T \widetilde{x} + b, \text{ with } \widetilde{\mathcal{W}_{:,i}} = \frac{\mathcal{W}_{:,i}}{\left\|\mathcal{W}_{:,i}\right\|_2}, i \in C \text{ and } \widetilde{x} = \frac{x}{\left\|x\right\|_2}, \tag{3}$$

where τ is a temperature factor and set as 20 in experiments. The normalized linear layer reduces the scale variance of features and weights of different categories, thus improves the performance of tail classes. Different from τ-norm [11] that only normalizes the weights at test time, our normalization is applied to both weights and features during training and testing. The combination of normalized linear layer and softmax shares a similar form of cosine softmax [15,20].

To further mitigate the extreme imbalance between background category and large vocabulary foreground categories, we adopt an objectness branch to predict objectness scores. This branch also adopts normalized linear layer, and is trained by cross-entropy loss.

During inference, both the classification score of various categories $score^{class}$ and score of objectness $score^{objectness}$ are activated with a softmax function. The final detection score $score_{det}$ for category i of a bounding box is

$$score_i^{det} = score_i^{class} * score^{objectness}. \tag{4}$$

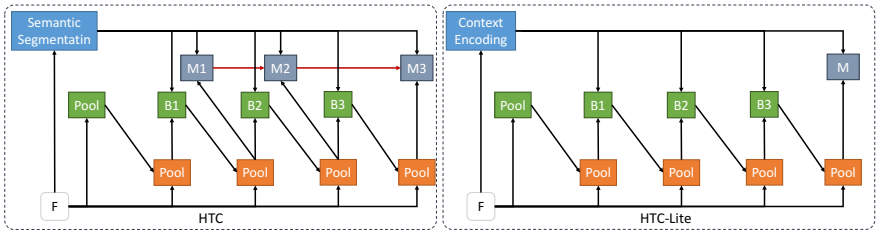

Fig. 1. The comparison of HTC and HTC-Lite.

Table 1. Performance comparison of Mask R-CNN with Cross-Entropy Loss (CE), Equalization Loss (EQL) and Seesaw Loss on LVIS *val* split trained with random sampler by 1x schedule.

Sampler	Loss	Bbox AP	Mask AP	AP_r	AP_c	AP_f
Random	CE	16.9	16.0	0.0	12.3	27.3
Random	EQL	19.3 (+2.4)	18.4 (+2.4)	1.8 (+1.8)	17.1 (+4.8)	27.1
Random	Seesaw	**24.3 (+7.4)**	**23.3 (+7.3)**	**13.0 (+13.0)**	**22.9 (+10.6)**	**28.2**

1.2 HTC-Lite

We propose HTC-Lite, a light-weight version of Hybrid Task Cascade (HTC) [3], to accelerate the training and inference speed while maintaining similar performance. As shown in Figure 1, the modification are in two folds: replacing the semantic segmentation branch by a global context encoding branch and reducing mask heads.

Context Encoding Branch. Since semantic segmentation annotations are unavialable for LVIS dataset, we replace the semantic segmentation branch by a global context encoder [22] trained by a semantic encoding loss. The context encoder applies convolution layers and global average pooling to obtain a vector of a image for multi-label prediction. This vector is also added to the RoI features used by box heads and mask heads.

Reduced Mask Heads. To further reduce the cost of instance segmentation, HTC-Lite only keeps one mask head in the last stage, which also spares the original interleaved information passing.

2 Experiments

Experimental Setting. We perform experiments on the LVIS v1 benchmark [7]. We use the *train* split for training and report the performance on the *val* split for ablation study. No external data and annotations are adopted except for standard ImageNet-1k [17] classification dataset for pre-training the backbone. We adopts mmdetection[4] as the codebase. Model ensemble is **not** adopted in our challenge entry.

2.1 Ablation Study of Seesaw Loss

We verify the effectiveness of Seesaw Loss on a Mask R-CNN with ResNet-50-FPN [13] Backbone, trained with multi-scale training and random sampler for 1x training schedule. We also compare Seesaw Loss with Equalization Loss (EQL)[19], the winning method in LVIS challenge 2019, to show the advantages of Seesaw Loss. As shown in Table 1, Seesaw loss significantly improves the baseline performance and surpasses EQL, especially on rare and common classes. The remarkable improvements on AP_r and AP_c validates the effectiveness of Seesaw Loss for long-tailed instance segmentation.

Table 2. Step by step results of our entry on LVIS v1 *val* split.

Modification	Schedule	Bbox AP	Mask AP	AP_r	AP_c	AP_f
Mask R-CNN	2x	20.1	18.7	1.0	16.1	29.4
+ SyncBN	2x	20.2 (+0.1)	18.9 (+0.2)	0.7	16.0	30.3
+ CARAFE Upsample	2x	20.4 (+0.2)	19.4 (+0.5)	0.7	16.5	30.9
+ HTC-Lite	2x	23.6 (+3.2)	21.9 (+2.5)	1.1	19.8	33.5
+ TSD	2x	25.5 (+1.9)	23.5 (+1.6)	2.3	22.3	34.0
+ Mask scoring	2x	25.6 (+0.1)	23.9 (+0.4)	2.8	22.4	35.0
+ Training-time augmentaion	45e	28.1 (+2.5)	26.5 (+2.6)	3.6	25.7	37.4
+ Better neck	45e	29.1 (+1.0)	27.0 (+0.5)	3.5	25.8	38.6
+ Better backbone	45e	32.1 (+3.0)	29.9 (+2.9)	4.2	29.4	41.8
+ **Seesaw Loss**	45e	39.8 (**+7.7**)	36.8 (**+6.9**)	25.5	35.6	42.9
+ Finetuning	1x	40.6 (+0.8)	37.3 (+0.5)	26.4	36.3	43.1
+ Test-time augmentation	-	**41.5** (+0.9)	**38.8** (+1.5)	**26.4**	**38.3**	**44.9**

2.2 Step by Step Results

Baseline. The baseline model is Mask R-CNN [8] using ResNet-50-FPN [13], trained with multi-scale training and random data sampler by 2x schedule [4].
SyncBN. We use SyncBN [14,16] in the backbone and heads.
CARAFE Upsample. CARAFE [21] is used for upsampling in the mask head.
HTC-Lite. We use HTC-Lite as described in Section 1.2.
TSD. TSD [18] is used to replace the box heads in all three stages in HTC-Lite.
Mask Scoring. We further use the mask IoU head [10] to improve mask results.
Training Time Augmentation. We train the model with stronger augmentations with 45 epochs. The learning rate is decreased by 0.1 at 30 and 40 epochs. We randomly resize the image with its longer edge in range of 768 to 1792 pixels. And then we randomly crop the image to size of 1280×1280 after adopting instaboost augmentation [6].
Better Neck. We replace the neck archtecture by an enhanced version of Feature Pyramid Grids (FPG) [2]. The enhanced FPG uses deformable convolution v2 (DCNv2) [24] after feature upsampling, and a downsampler version of CARAFE [21] for feature downsampling.
Better Backbone. We use ResNeSt-200 [23] with DCNv2 [24].
Seesaw Loss. We apply the proposed Seesaw Loss to classification branches of the TSD box head, in all cascading stages. Furthermore, we remove the original progressive constraint (PC) loss on classification branches in TSD.
Finetuning with Repeat Factor Sampling. After obtaining the model with Seesaw Loss trained by a random sampler, we freeze all components in the original model. Then we finetune a new classification branch for each cascading stage on the fixed model using repeat factor sampler [7] by 1x schedule. During inference, the classification scores of original classification branches and the scores of finetuned classification branches are averaged to get the final scores.
Test Time Augmentation. We adopt multi-scale testing with horizontal flipping. Specifically, images scales are 1200, 1400, 1600, 1800 and 2000 pixels.
Final Performance on Test-dev. After adding the abovementioned components step by step, we finally achieve **38.8% AP** on the *val* split and **38.92% AP** on the *test-dev* split.

References

1. Cao, K., Wei, C., Gaidon, A., Arechiga, N., Ma, T.: Learning imbalanced datasets with label-distribution-aware margin loss. In: Advances in Neural Information Processing Systems (2019)
2. Chen, K., Cao, Y., Loy, C.C., Lin, D., Feichtenhofer, C.: Feature pyramid grids (2020)
3. Chen, K., Pang, J., Wang, J., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Shi, J., Ouyang, W., Loy, C.C., Lin, D.: Hybrid task cascade for instance segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
4. 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)
5. Cui, Y., Jia, M., Lin, T.Y., Song, Y., Belongie, S.: Class-balanced loss based on effective number of samples. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
6. Fang, H.S., Sun, J., Wang, R., Gou, M., Li, Y.L., Lu, C.: Instaboost: Boosting instance segmentation via probability map guided copy-pasting. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 682–691 (2019)
7. Gupta, A., Dollar, P., Girshick, R.: LVIS: A dataset for large vocabulary instance segmentation. In: CVPR (2019)
8. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask r-cnn. 2017 IEEE International Conference on Computer Vision (ICCV) (Oct 2017)
9. Hu, X., Jiang, Y., Tang, K., Chen, J., Miao, C., Zhang, H.: Learning to segment the tail. In: CVPR (2020)
10. Huang, Z., Huang, L., Gong, Y., Huang, C., Wang, X.: Mask scoring r-cnn. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
11. Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., Kalantidis, Y.: Decoupling representation and classifier for long-tailed recognition. In: ICLR (2020)
12. Li, Y., Wang, T., Kang, B., Tang, S., Wang, C., Li, J., Feng, J.: Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10991–11000 (2020)
13. Lin, T., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. In: CVPR (2017)
14. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: CVPR (2018)
15. Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., Yu, S.X.: Large-scale long-tailed recognition in an open world. In: CVPR (2019)
16. Peng, C., Xiao, T., Li, Z., Jiang, Y., Zhang, X., Jia, K., Yu, G., Sun, J.: MegDet: A large mini-batch object detector. CVPR (2018)
17. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. IJCV (2015)
18. Song, G., Liu, Y., Wang, X.: Revisiting the sibling head in object detector. CVPR (2020)
19. Tan, J., Wang, C., Li, B., Li, Q., Ouyang, W., Yin, C., Yan, J.: Equalization loss for long-tailed object recognition. In: CVPR (2020)
20. Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., Liu, W.: Cosface: Large margin cosine loss for deep face recognition. CVPR (2018)
21. Wang, J., Chen, K., Xu, R., Liu, Z., Loy, C.C., Lin, D.: Carafe: Content-aware reassembly of features. In: The IEEE International Conference on Computer Vision (ICCV) (October 2019)

22. Zhang, H., Dana, K., Shi, J., Zhang, Z., Wang, X., Tyagi, A., Agrawal, A.: Context encoding for semantic segmentation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
23. Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Zhang, Z., Lin, H., Sun, Y., He, T., Muller, J., Manmatha, R., Li, M., Smola, A.: Resnest: Split-attention networks. arXiv preprint arXiv:2004.08955 (2020)
24. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets v2: More deformable, better results. In: CVPR (2019)