LVIS Challenge Track Technical Report: Distribution Balanced and Boundary Refinement for Large Vocabulary Instance Segmentation

WeiFu Fu[*], **CongChong Nie**[*], **Ting Sun**, **Jun Liu**, **TianLiang Zhang**, and **Yong Liu**

Tencent YouTu Lab

ryanwfu@tencent.com, nickncc2122@gmail.com, yolandasun@tencent.com,
junsenselee@gmail.com, linuszhang@tencent.com, choasliu@tencent.com

Abstract

This report introduces the technical details of the team FuXi-Fresher for LVIS Challenge 2021. Our method focuses on the problem in following two aspects: the long-tail distribution and the segmentation quality of mask and boundary. Based on the advanced HTC instance segmentation algorithm, we connect transformer backbone(Swin-L) through composite connections inspired by CBNetv2 to enhance the baseline results. To alleviate the problem of long-tail distribution, we design a Distribution Balanced method which includes dataset balanced and loss function balaced modules. Further, we use a Mask and Boundary Refinement method composed with mask scoring and refine-mask algorithms to improve the segmentation quality. In addition, we are pleasantly surprised to find that early stopping combined with EMA method can achieve a great improvement. Finally, by using multi-scale testing and increasing the upper limit of the number of objects detected per image, we achieved more than 45.4% boundary AP on the val set of LVIS Challenge 2021. On the test data of LVIS Challenge 2021, we rank 1st and achieve 48.1% AP. Notably, our APr 47.5% is very closed to the APf 48.0%.

1 Introduction

LVIS [1] is a new benchmark dataset for large vocabulary instance segmentation. It contains about 2 million high-quality instance segmentation annotations in more than 1,000 categories. Compared with the COCO [2] dataset, the LVIS dataset has a long-tail distribution which is more similar to the real world. According to the statistics, the rare categories only accounts for about 1/3 of the total number of categories and 0.41% of the total number of instances. Thus, the unbalanced data increases the difficulty of training which is the first challenge in LVIS workshop. In addition, more and more attention has been paid to more precise segmentation result. Therefore, the boundary AP evaluation metric [3] is proposed and puts forward higher requirements on the mask prediction quality. However, many algorithms fail to well segment the boundary of mask. It is the other challenge in LVIS workshop.

To solve these challenges, our solution proposed two algorithmic modules: (1) Distribution Balanced Algorithm: including RFS, balanced copy-paste and balanced Mosaic augmentation and Seesaw loss function to deal with long-tail distribution; (2) Mask and Boundary Refinement Algorithm: including re-scoring the mask prediction and refinemask methods to optimize the quality of mask and boundary. Overall, based on effective data augmentation strategies, an optimized loss function and a stronger backbone with the two-stage HTC [4] model, we improve the performance on long-tail distribution

---

[*]indicates equal contribution.

problem. Further, refinemask [5] gradually integrate fine-grained features and refine the mask, and mask scoring re-scoring the mask prediction. The details will be described in Section 2.

In addition, we utilized early stopping combined with EMA method in learning based on our the observation of the experiment. The method has a great improvement and is robust. The details will be described in Section 3.2.

Overall, On the test data of LVIS Challenge 2021, we achieve 48.1% AP. Notably, our APr 47.5% is very closed to the APf 48.0%.

2 Our Solution

2.1 Strong Baseline

(1) Hybrid Task Cascade (HTC): HTC is a classic yet powerful cascade architecture for instance segmentation task. It interweaves detection and segmentation branches for a joint multi-stage processing and progressively learns more discriminative features in each stage. In our solution, we removed the semantic head since it requires extra stuff segmentation annotations.

(2) Swin Transformer [6] and CBNetV2 [7] Backbone: Vision transformer has made a great progress in many visual tasks recently. Thus, we use Swin Transformer as our backbone. Swin Transformer proposed an efficient window attention module in a hierarchical feature architecture. It has a linear computational complexity with respect to the input image size. In this work, we use Swin-L network as our basic backbone which is pretained by Imagenet-22k dataset. Furthermore, inspired by CBNetv2 algorithm, we groups two identical Swin-L networks through composite connections as the final backbone to enhance the performance.

2.2 Distribution Balanced

(1) Repeat factor sampling (RFS) [1]: Taking into account the serious imbalance of the LVIS dataset, we use RFS to increase the sampling rate of the images containing the tail categories.

(2) Balanced Copy-Paste: Considering that RFS is an image-level re-sampling technology, we further adopt copy-paste [8] data augmentation method, which can be considered as an instance-level re-sampling technology. It can increase the sampling probability of the tail categories instances, and can avoid the mixing of the head categories instances caused by RFS.

(3) Balanced Mosaic: The balanced mosaic [9] strategy randomly samples among four training images for stitching based on the RFS re-sampling dataset. It can significantly increase the number of instances in each image and enrich the background. We note that the scale of instance size will be reduced relatively in the generated image which will result in some distribution difference between the train set and the test set. Therefore, We use a probability of 0.5 to randomly choose training images to do balanced mosaic for avoiding distribution shift.

2.3 Loss Function

Seesaw Loss: Seesaw Loss [10] uses mitigation factor and compensation factor to dynamically suppress excessive negative sample gradients on the tail category, while supplementing the penalty for misclassified samples to avoid false positives.

2.4 Mask and Boundary refinement

Although our strong baseline has achieved a great improvement, the instance mask generated by the HTC model is still very coarse due to the pooling process and the 28x28 masks representation size. Therefore, we use refinemask to replace the mask head of the last stage of HTC to obtain a higher quality mask prediction. At the same time, we also used mask scoring to re-scoring the final mask to further improve the mask metric.

(1) Mask Scoring: Directly using classification confidence as the mask score will result in misregistration of the mask score and the mask quality. Mask scoring [11] can automatically learn the mask quality instead of relying on the classification confidence of the bounding box.

(2) RefineMask: The refinemask [5] gradually integrates fine-grained features and designs boundary supervision to obtain high-quality masks. Referring to the refinemask of the maskRCNN version, we initially replaced the three mask heads of HTC with refinemask heads. However, this will result in a significant increase in GPU memory and training time. In order to balance the training time and accuracy, we only replace the last mask head with a refinemask head. In addition, we set the loss weight of bbox head to 2.0 to balance the weight between bbox head and mask head.

3   Experiments

3.1   Implementation Details

In our experiments, we use HTC without semantic branch as our baseline. The batch size is set to 16. The learning rate is set to 0.0001 and 0.00005 corresponding to single swin-L backbone and CBNetV2 backbone. We train all models for total 20 epochs by default, with learning rate divided by 10 at the 16th and 19th epoch. All models are initialized with ImageNet-22k pre-trained model.

3.2   Experimental Results

3.2.1   Distribution Balanced and Strong Baseline

We use RFS and seesaw Loss to deal with the long tail problem from the perspective of data re-sampling and loss function. Note that, we directly adopt Swin-L as the basic backbone because some methods are effective on a simple backbone but do not work on a stronger backbone. The basic experimental results in Table 1 achieved 39.1% mask AP (**named Weak Baseline**), which directly surpasses the last year's third place. Subsequently, experiments with balanced copy-paste and balanced mosaic achieves mask AP improvements of 0.7% and 1.0%, respectively. In addition, the improvement of AP_r is more significant, achieving an improvement of 3.9% and 1.5% respectively. After adopting a stronger CBNetV2 backbone, mask AP has been further improved to 43.1% (**named Strong Baseline**), and AP_r, AP_c, and AP_f have also been greatly improved.

RFS	SL	Swin-L	B-C	B-M	CBV2	mask AP(%)	$AP_r(\%)$	$AP_c(\%)$	$AP_f(\%)$
✓	✓	✓				39.1	27.9	39.2	43.7
✓	✓	✓	✓			39.8	31.8	40.2	42.8
✓	✓	✓	✓	✓		40.8	33.3	40.5	44.3
✓	✓	✓	✓	✓	✓	43.1	34.3	43.1	47.0

Table 1: Ablation studies on LVIS v1.0 val set. Models are HTC w/o semantic branch. RFS: Repeat Factor Sampling. SL: Seesaw Loss. B-C: Balanced Copypaste. B-M: Balanced Mosaic. CBV2: CBNetV2. We also used multi-scale training, random flip, crop and color jitter in all experiments. It is worth noting that the experiments using the balanced mosaic adopt the short-side zoom range of (640, 1400), and the other experiments use the zoom range of (400, 1400), which is to avoid generating too many extremely small objects.

3.2.2   Ablation of Refinemask

We only conduct the refinemask ablation experiment on Week Baseline due to time constraints. As shown in Table 2, refinemask can lead to +1.1% in mask AP. Meanwhile, the metrics of AP_r, AP_c and AP_f also has an improvement. Thus, we infer that refinemask can also improve the performance on Strong Baseline. It's worth noting that we only replace the last mask head with a refinemask head in HTC.

3.2.3   Early Stopping and EMA

As shown in Figure 1 (a), we visualize the AP curve of the strong baseline on LVIS validation set. It can be seen that the curves of AP_r and AP_f have a large contrast, reaching their peaks at 6th epoch

Method	epoch	mask AP	$AP_r(\%)$	$AP_c(\%)$	$AP_f(\%)$
Weak Baseline	20	39.1	27.9	39.2	43.7
+refinemask	20	40.2	29.1	40.3	44.9

Table 2: Refinemask ablation studies on LVIS v1.0 val set based on a Weak Baseline.

and 20th epoch, respectively. However, the mask AP of 6th epoch and 20th epoch are very similar. We think this is caused by the long tail distribution problem.

In addition, considering the large fluctuation in the performance of the strong baseline model, we also adopt the Exponential Moving Average (EMA) [12] training strategy to smooth the model weights. As shown in Figure 1 (b), we visualize the AP curve of the model in row 2 of Table 3. It can be seen that the mask AP, AP_r and AP_c at 6th epoch both obtain a remarkable increase compared with 20th epoch. Above all, the gap between AP_r, AP_c and AP_f is also significantly resolved. Therefore, we finally adopted the early stopping and EMA strategy in subsequent models.
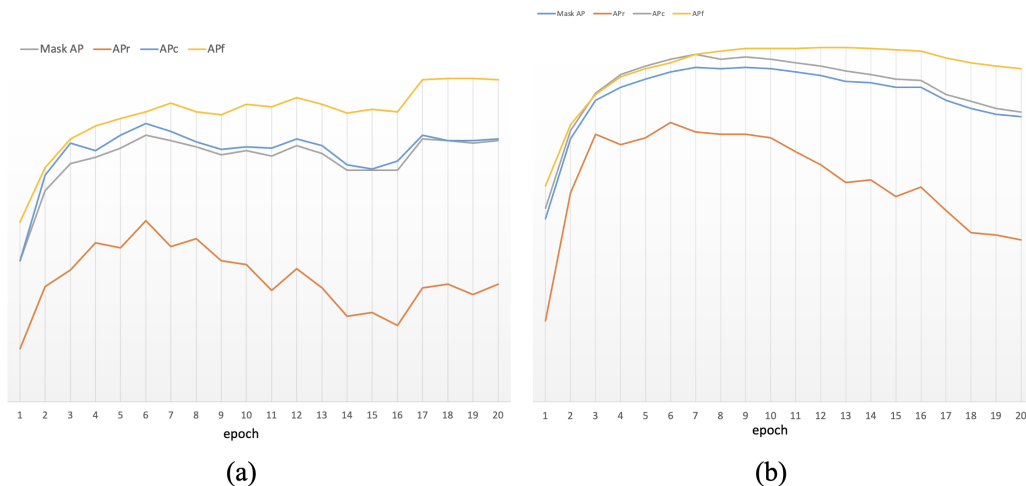

Figure 1: AP curve on the LVIS validation set. (a) AP curve of strong baseline model in row 1 of Table 3. (b) AP curve of model in row 2 of Table 3.

3.2.4 Mask and Boundary refinement

On the basis of strong baseline, we report the experimental results of refinemask and mask scoring in Table 3. In particular, we use EMA, early stopping and FP32 in row2 and row3 of Table 3 to further improve accuracy. It can be seen that the combination of EMA, early stopping, FP32 and refinemask can improves the mask AP by 5%. After further integrating mask scoring, mask AP obtains a higher result of 49.2 mask AP on the val set.

EMA	early stop	FP32	refinemask	mask scoring	Mask AP(%)	$AP_r(\%)$	$AP_c(\%)$	$AP_f(\%)$
					43.1	34.3	43.1	47.0
✓	✓	✓	✓		48.1	43.8	49.2	48.9.
✓	✓	✓	✓	✓	49.2	45.4	50.0	50.0.

Table 3: Mask scoring and refinemask ablation studies on LVIS v1.0 val set. Both mask scoring and refinemask head are only added to the last stage of HTC.

4 Final Results

4.1 Test Time Augment (TTA)

As shown in Table 4, we use standard multi-scale testing, which adopts multiple test resolutions ((1600, 1000), (1600, 1400), (1800, 1200), (1800, 1600)) and flip strategy.

TTA	Mask AP(%)	Boundary AP(%)	AP_r(%)	AP_c(%)	AP_f(%)
	49.2	44.1	40.4	45.3	44.4
✓	50.4	45.4	41.1	46.5	45.9

Table 4: Test time augmentation ablation studies on LVIS v1.0 val set.

4.2 More Detections Per Image

In this year's competition rules, there is no limit on the number of detections per image. As shown in Table 5, we increase the number of detections per images to 1000 finally.

max_per_img	Boundary AP(%)	AP_r(%)	AP_c(%)	AP_f(%)
300	44.1	40.4	45.3	44.4
500	44.9	42.6	45.9	44.8
800	45.2	43.4	46.2	44.9
1000	45.3	43.5	46.3	44.9

Table 5: More detections per image ablation studies on LVIS v1.0 val set.

4.3 Test Challenge Result

Our final model applies TTA and increases the number of objects detected per image to 1000, which will exceed 45.4% boundary AP on LVIS v1.0 val set. Due to time constraints, we did not test its results on the LVIS v1.0 val set, but on LVIS v1.0 Test Challenge, we used both in the final submission. Our final result is shown in Table 6. It can be seen that our result has almost equal results on AP_r, AP_c and AP_f, which confirms the effectiveness of our solution on the long-tail distribution problem.

Set	TTA	max_per_img	$AP\dagger$(%)	$AP_r\dagger$(%)	$AP_c\dagger$(%)	$AP_f\dagger$(%)
Val	✓	300	45.4	41.1	46.5	45.9
Val		1000	45.3	43.5	46.3	44.9
Test challenge	✓	1000	48.1	47.5	48.5	48.0

Table 6: Final Result on LVIS v1.0 Test Challenge. AP†: refers to the "boundary, fixed AP [13]" used in the LVIS challenge 2021.

References

[1] A. Gupta, P. Dollar, and R. Girshick, "Lvis: A dataset for large vocabulary instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5356–5364, 2019.

[2] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014.

[3] B. Cheng, R. Girshick, P. Dollár, A. C. Berg, and A. Kirillov, "Boundary iou: Improving object-centric image segmentation evaluation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15334–15342, 2021.

[4] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang, *et al.*, "Hybrid task cascade for instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4974–4983, 2019.

[5] G. Zhang, X. Lu, J. Tan, J. Li, Z. Zhang, Q. Li, and X. Hu, "Refinemask: Towards high-quality instance segmentation with fine-grained features," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6861–6869, 2021.

[6] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *arXiv preprint arXiv:2103.14030*, 2021.

[7] T. Liang, X. Chu, Y. Liu, Y. Wang, Z. Tang, W. Chu, J. Chen, and H. Ling, "Cbnetv2: A composite backbone network architecture for object detection," *arXiv preprint arXiv:2107.00420*, 2021.

[8] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2918–2928, 2021.

[9] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. L. YOLO, "Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[10] J. Wang, W. Zhang, Y. Zang, Y. Cao, J. Pang, T. Gong, K. Chen, Z. Liu, C. C. Loy, and D. Lin, "Seesaw loss for long-tailed instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9695–9704, 2021.

[11] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring r-cnn," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6409–6418, 2019.

[12] https://www.tensorflow.org/api_docs/python/tf/train/ExponentialMovingAverage.

[13] A. Dave, P. Dollár, D. Ramanan, A. Kirillov, and R. Girshick, "Evaluating large-vocabulary object detectors: The devil is in the details," *arXiv preprint arXiv:2102.01066*, 2021.