Joint COCO and LVIS workshop at ECCV 2020: LVIS Challenge Track Technical Report: Balanced Mosaic and Double Classifier for Large Vocabulary Instance Segmentation

Lei Chen*, Qiang Zhou*, Wei Li*, Zhibin Wang, and Hao Li

Alibaba Group
{fanjiang.cl, jianchong.zq, lw263154, zhibin.waz, lihao.lh}@alibaba-inc.com

Abstract. With the release of the large vocabulary dataset LVIS [5], the long-tail instance segmentation has received more and more attention. In this work, we propose double classifeir and double sampler, which achieve 0.8% AP gain for all categories and 8.3% AP gain for rare categories, compared to the strong baseline. Furthermore, we combine mosaic augmentation [1] with re-sample method to build a more balanced dataset, meanwhile we introduce richer transforms to achieve better generalization. we name this method as balanced mosaic. Balanced mosaic further achieves 3.9% AP gain (35.4 vs 36.8). Finaly, with multi scale test, we achieves 39.2 mask AP on test set of the LVIS Challenge 2020.

1 Introduction

LVIS dataset has a large number of categories and some categories' images are significantly less than others. The long tail nature of lvis dataset poses a huge challenge to model training. Some existing work, such as classifier retraining (cRT) [9], classification calibration [12], balanced group softmax(BGS) [10] have shown that decoupling feature learning and classifier learning can help improve model's performance on long-tail categories. In addition to decoupling training, in this work, we also propose two effective solutions to improve the performance on the rare categories. First, we propose balanced mosaic augmentation to make training samples more balanced and diverse, which will be described in Section 2. Second, we design a double classifier structure to obtain high precision on all categories. One of the classifiers is retrained with repeat factor sampling and the other classifier is retrained with class-away sampling. The detail will be described in Section 3.

* indicates equal contribution

2 Balanced Mosaic

Mosaic augmentation was first introduced by YoloV4 [1]. It is a new data augmentation method that mixes 4 training images, which allows detection of objects outside their normal context. In addition, batch normalization would calculate activation statistics from 4 different images on each layer, thus significantly reduces the need for large mini-batch size. In this work, we use mosaic augmentation to solve the long tail instance segmentation problem. We sample images containing tail categories with a higher probability and head categories with a lower probability, So the mosaic images are more balanced. The psuedo-code of this algorithm can be found in Algorithm 1. It mainly consists of three steps, First, we compute sampling probability for each image in the dataset. There are several ways to achieve this purpose, here we obtain the sampling probability of image i by normalizing the image level repeat factor r_i, which has detailed description in repeat factor sampling(RFS) [5]. Second, pre-transforms are applied to each sampled image, we simply adopt resize operation in this step. Third, post-transforms are applied to the mosaic images, including crop, color jitter and flip operations, which can reduce overfitting.

Algorithm 1 Balanced mosaic

Input: x_i, $X = \{x_0, x_1, ..., x_n\}$
Output: y_i
Step1 Compute sampling probability:

$$p_i = r_i / \sum r_i$$

$$x^m = \{x_i + \text{random}(X, 3, p = p)\}$$

Step2 Apply pre-transform:
a) random a scale s accoring to the shape of x_0^m
b) for x_i^m in x^m

$$x_i^m = \text{Resize}(x_i^m, s)$$

Step3 Do mosaic:

$$y_i = \text{Mosaic}(x^m)$$

Step4 Apply post-transform:

$$y_i = \text{ColorJitter}(\text{Flip}(\text{Crop}(y_i)))$$

3 Double Classifier

Inspired by [9], we divide the model training process into two phases: representation learning and box classifier retraining. In the representation learning phase, we observed that choosing repeat factor sampling [6] instead of instance-away sampling (uniform sampling) will eventually obtain better results. During the box classifier retraining phase, we found that one box classifier with only

one sampling strategy is not able to achieve good performance on all categories. For example, applying class-away sampling will achieve highest precision on rare categories, but will perform poorly on common and frequent categories. And applying repeat factor sampling will achieve higher precision on common and frequent categories, while the precision on rare categories will be slightly lower.

Therefore, we propose double classifier and double sampler. Specifically, in box classifier retraining phase, we retrain two box classifiers, one with repeat factor sampling and one with class-away sampling. When conduct class-away sampling, for each epoch, we resample 5 images for each category, thus each epoch will contain 1203×5 images and these images are different in different epoches. Since the two box classifier heads share all parameters, during inference, we simply average the predictions of the two box classifier heads to get the final box scores without reducing inference speed of the model.

4 Experiments

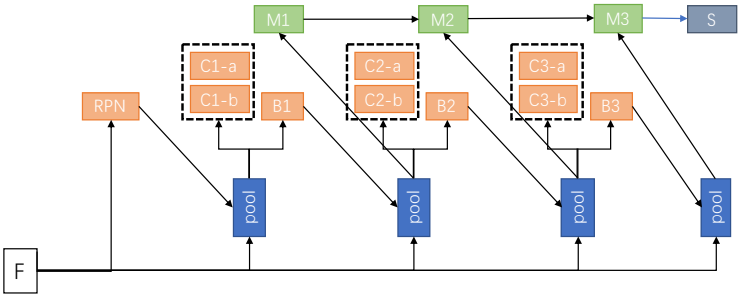

Fig. 1. Final challenge model based on HTC[3]

4.1 Challenge model and Single-scale Testing

Challenge Model As shown in Figure 1, our challenge model is based on HTC, with backbone Senet154 [7], FPN, double head [13] and maskiou head [8] (to save memory, we only add maskiou head on the last stage). Deformable convolution [4] is adopted in stage3, stage4 and stage5 in the backbone. We also use 3-stage Recursive Feature Pyramid [11] to improve the performance of our model. Unless otherwise stated, the models are trained on LVIS-v1.0 training set and evaluated on LVIS-v1.0 validation set for mask prediction tasks. All the models are trained with SGD using 3 Nvidia V100 GPU. Super-parameters, such as momentum and batch size are set to 0.9 and 24 respectively. We update the learning rate in the 20th/24th/26th/27th/28th epoch to 3e-2/3e-3/3e-4/3e-5/3e-6 respectively. We use multi-scale training and the shorter size of images are random sampled from 600 to 1000, and the longer edge is set to 1400. Due to time constraints, we run all the following experiments only once.

Table 1. Ablation study results of our final challenge model on LVIS-v1.0 *val* set. Images' short sizes are set to 800.

Model	AP	AP_r	AP_c	AP_f
Representation learning	33.4	17.3	33.2	40.6
+BGS[10] retraining	35.1	22.7	34.8	40.7
+double classifier	35.4	24.6	34.9	40.7
+balanced mosaic	36.8	27.7	36.7	41.0
+coco2017_unlabeled	36.9	29.2	36.4	40.9

Representation Learning We use repeat factor sampling [6] during the represenation learning stage, and get 33.4% AP in LVIS-v1.0 val set, as thown in Table 1

BGS Retraining Balanced group softmax [10] split all categories into several disjoint groups and perform the softmax operation separately, such that only classes with similar numbers of training instances are competing with each other within each group. In the box classifier retraining phase, we replace all three box classifers (C1-a, C2-a, C3-a in Figure 1) with balanced group softmax, and obtain a higher result of 35.1% AP on the val set.

Double Classifier In the box classifer retraining phase, we also use class-away sampling to retrain three box classifers (C1-b, C2-b, C3-b in Figure 1) which performs better in rare categories. During inference, the box scores of each stage S_i is obtained by averaging the prediction results of the two classifiers, i.e., $S_i = (S_{Ci-a} + S_{Ci-b})\ /\ 2$.

Balanced Mosaic Due to time constraints, we only use balanced mosaic during box classifier retraining, which further imporved the result from 35.4% to 36.8% AP, which proves that the balanced mosaic method is a very effective method. We believe that if the balanced mosaic method is used in the representation learning phase, the result may be better.

Unlabeled Data We use the trained model as the teacher model to generate pseudo labels on the coco2017_unlabeled datas (including bounding boxes, class labels and masks) with a high score threshold of 0.8. Since frequent categories already have enough training samples, we only preserve images whose pseudo-labels including at least one rare category. We use pseudo label data in the box classifier retraining phase, together with LVIS-v1.0 training set. In addition, we apply strong data augmentations (including blur, color jitter, cutout) on unlabeled data and apply dropout in the box classifier head.

4.2 Multi-scale Testing

We perform multi-scale testing on both bounding box and segmentation results. Specifically, based on the obtained bounding box merging result, we crop mask proposals on different inferring scales and send them to mask head. The voting results from different inferring scales are employed. The testing scales are set

to (1200, 800), (1500, 1000), (1800, 1200), (2100, 1400), (2400, 1600). No other augmentation is used except horizontal flipping. Additionly, noticed that instances of different sizes have performance gaps at different test scales, we adopt scale-adaptive merging strategy to collect the best results in different test scales. Soft-nms [2] is also used to obtain better performance and the IoU threshold is set to 0.5. Testing results with different strategies on *val* are shown in Table 2 and the comparison between official baseline based on ResNeXt-101-32x8 and our method is shown in Table 3.

Table 2. Experiment results on *val* with different testing strategies.

Model	AP	AP_r	AP_c	AP_f
Challenge Baseline	36.9	29.2	36.4	40.9
+Multi-scale box	39.1	29.8	38.4	43.9
+Soft-nms	39.2	30.2	38.5	44.1
+Scale-adaptive	39.4	30.3	38.7	44.2
+Multi-scale mask	**39.8**	**30.4**	**39.1**	**44.7**

Table 3. Comparision between official baseline and our method on *val* and *test dev*.

		AP	AP_r	AP_c	AP_f
Baseline	val	27.26	19.47	26.13	31.95
Ours	val	**39.8**	**30.4**	**39.1**	**44.7**
Baseline	test_dev	26.86	20.41	24.90	31.97
Ours	test_dev	**39.21**	**29.72**	**37.79**	**45.08**

References

1. Bochkovskiy, A., Wang, C., Liao, H.M.: Yolov4: Optimal speed and accuracy of object detection. CoRR **abs/2004.10934** (2020), https://arxiv.org/abs/2004.10934
2. Bodla, N., Singh, B., Chellappa, R., Davis, L.S.: Soft-nms–improving object detection with one line of code. In: Proceedings of the IEEE international conference on computer vision. pp. 5561–5569 (2017)
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 (2019)
4. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks (2017)
5. Gupta, A., Dollár, P., Girshick, R.B.: LVIS: A dataset for large vocabulary instance segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition,

CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. pp. 5356–5364. Computer Vision Foundation / IEEE (2019). https://doi.org/10.1109/CVPR.2019.00550
6. Gupta, A., Dollár, P., Girshick, R.: Lvis: A dataset for large vocabulary instance segmentation (2019)
7. Hu, J., Shen, L., Albanie, S., Sun, G., Wu, E.: Squeeze-and-excitation networks (2017)
8. Huang, Z., Huang, L., Gong, Y., Huang, C., Wang, X.: Mask scoring r-cnn (2019)
9. Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., Kalantidis, Y.: Decoupling representation and classifier for long-tailed recognition. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=r1gRTCVFvB
10. 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. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (Jun 2020). https://doi.org/10.1109/cvpr42600.2020.01100, http://dx.doi.org/10.1109/cvpr42600.2020.01100
11. Qiao, S., Chen, L.C., Yuille, A.: Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution (2020)
12. Wang, T., Li, Y., Kang, B., Li, J., Liew, J.H., Tang, S., Hoi, S.C.H., Feng, J.: Classification calibration for long-tail instance segmentation. CoRR **abs/1910.13081** (2019), http://arxiv.org/abs/1910.13081
13. Wu, Y., Chen, Y., Yuan, L., Liu, Z., Wang, L., Li, H., Fu, Y.: Rethinking classification and localization for object detection (2019)