Joint COCO and LVIS workshop at ECCV 2020: LVIS Challenge Track Technical Report: Multi-expert heads of Long-tailed Instance Segmentation

Wan-Cyuan Fan[2], Cheng-Yao Hong[1], Yen-Chi Hsu[1,2], and Tynh-Luh Liu[1]

[1] Institute of Information Science, Academia Sinica, Taipei, Taiwan
{sensible yenchi liutyng}@iis.sinica.edu.tw
[2] Department of Computer Science & Information Engineering, National Taiwan University christine5200312@gmail.com

Abstract. The Large Vocabulary Instance Segmentation (LVIS) dataset is a large-scale object detection task with thousands of categories, which causes severe challenges due to the extremely long-tailed data distributions. The common methods (re-weighting or re-sampling) which might damage the representation due to the data distribution is distorted. In this work, we proposed a multi expert detection head approach to alleviate the ordeals brought by imbalanced numbers between the foregrounds without damaging the representation. The result of the proposed model on the validation and testing-dev set achieves 34.1 and 33.7 mAP, respectively, of the European Conference on Computer Vision challenge (ECCV) 2020.

1 Introduction

Large Vocabulary Instance Segmentation (LVIS) [5] contains instance segmentation masks for more than 1000 object categories with a challenging long tail of rare objects. It divides the objects into rare, common, and frequent level. To solve the long-tailed problem, [5] introduce a dynamic sampling factor that enables rare data to automatically appear several times during training. This will allow rare data to be more evenly distributed. However, the recent works [6, 11] point out that the image-wise re-sampling methods damage the representation. We propose the multi-expert detection head approach to learn the embedding for each level (rare, common, and frequent). This can alleviate the rare level to be divided into the background without damaging the representation. Also, we use the synthetic instance augmentation to balance the instances count of the dataset. After that, we use multi-scale image augmentation that enables the model to capture the size of the object more accurately. It also avoids the model that has difficulty capturing large or small objects. And lastly, following the [6, 7] we use two-phase training to fine-tune the classifier, thus further improving our model. Those methods improve the validation performance and testing results which outperform the baseline.

2 Approach

We propose two methods to solve the instance imbalance problem when training the instance classification. One is multi-expert heads and another is synthetic instance augmentation. Multi-expert heads let each head focus on their main task and make the self-ensemble with the expert way. Synthetic instance augmentation brings the balance between rare and frequency instances. With the advantages of both, the multi-stage architectures gain a significant improvement on the LVIS (v0.5/v1.0) dataset.

2.1 Multi-Expert Heads

Previous work. Given a multi-stage architecture model [1, 3] f, we have $f : x \longrightarrow \{o_1, ..., o_n\}$, where x means the input image and o_j is the output of each head. In instance segmentation task, $o_j = \{\text{cls}_j, \text{bbox}_j, \text{seg}_j\}$ where $\text{cls}_j, \text{bbox}_j$, and seg_j are instance classification, instance bounding box, and instance segmentation, respectively.

Expert Training. In here, we are going to make each head become an expert according to the instance frequency. In LVIS challenge, they split the instance frequency into rare, common, and frequency. Hence, following the setting of LVIS, our multi-stage architecture model becomes $f : x \longrightarrow \{o_r, o_c, o_f\}$ where o_r, o_c, and o_f are the outputs from the expert head with rare, common, and frequency, respectively. Since the challenge in LVIS is the instance classification, we design an expert loss function for $\text{cls}_r, \text{cls}_c$, and cls_f.

In the conventional classification approach, we have $\text{cls}_j = \mathbf{z}, \mathbf{z} \in \mathbb{R}^K$, where $\mathbf{z}$ means the logit with K categories. To make the logit become the probability of each category, we usually use the softmax activation function

$$p_i = \frac{e^{z_i}}{\sum_{i=1}^{K} e^{z_i}}. \tag{1}$$

Next, we try to let each head attends on the instance frequency, we split the categories into rare, common, and frequency sets with S_r, S_c, and S_f. Then, we have expert softmax

$$Ep_i = \frac{e^{z_i}}{\sum_{i \in S_E} e^{z_i}}, \tag{2}$$

where $E \in \{r, c, f\}$ means the expert set. Empirically, increasing the intersection over union (IoU) threshold will decrease the performance of rare classification but improve the performance of frequency. Hence, we assign the rare for the first head, common for the second head, and frequency for the third head.

There is a simple method to combine the general softmax and expert softmax on each head for calculating the losses. We take the union between general and

expert probability on the target category and then calculate the losses by cross-entropy:

$$\mathcal{L}_{\text{cls}} = -\sum_{i=1}^{K} \mathbb{I}_{c_i=c_t} \log\left(p_i\right) - \sum_{i \in S_E} \mathbb{I}_{c_i=c_t} \log\left(Ep_i\right) \tag{3}$$

$$= -\sum_{i \in S_E} \mathbb{I}_{c_i=c_t} \log\left(p_i \times Ep_i\right). \tag{4}$$

Expert Inference. At the inference time, multi-stage architecture [1, 3] use an average self-ensemble to make the final instance probability prediction. The prediction $\mathbf{p}$ is

$$\mathbf{p} = \frac{1}{n} \sum_{j=1}^{n} \sum_{k=1}^{K} p_{j,k} \mathbf{e}_k, \tag{5}$$

where $\mathbf{e}$ is the unit vector with $\mathbf{e}_k = 1, \mathbf{e}_{i \neq k} = 0$, index j means the number of head, and n usually is 3.

While training with the multi-expert heads approach, each head becomes an expert according to the instance frequency. Hence, we replace the average self-ensemble by expert self-ensemble. The instance probability prediction $\mathbf{p}$ becomes

$$\mathbf{p} = \sum_{k \in S_r} p_{r,k} \mathbf{e}_k + \sum_{k \in S_c} p_{c,k} \mathbf{e}_k + \sum_{k \in S_f} p_{f,k} \mathbf{e}_k \tag{6}$$

$$= \sum_{E \in \{r,c,f\}} \sum_{k \in S_E} p_{E,k} \mathbf{e}_k. \tag{7}$$

In colloquial terms, we combine the inference prediction from the first head, the second head, and the third head only with the rare set, common set, and frequency set, respectively.

2.2 Synthetic Instance Augmentation

Another approach is synthetic instance augmentation. The main idea is to create some synthetic images with lots of rare instances and some common and frequency instances. Note that there is no external data. All of the synthetic annotations are created from the training dataset.

To create the synthetic images, first, we randomly pick up an image from the training data which has no annotations. Second, randomly select some instances with the proportion rare:common:frequency= $a : b : c$, where $a > b > c$, and $a, b, c \in \mathbb{N}$. Finally, randomly paste the instances on the selected image for each of them is nonoverlapping. In our experiments, we have a, b, and c be $5, 3$, and 1.

Creating some synthetic images with more rare instances can alleviate the negative gradient problem for rare instances. In addition, it also avoids the overfitting of the classification on the rare instances rather than the repeat factor

sampling [5]. Since repeat factor sampling just re-sample the same images with the same annotations which means the background of the instances are still the same.

3 Experiments Details

We perform experiments on the LVIS dataset [5], which contains 1203 categories in release v1.0 and 1230 categories in release v0.5. The evaluation metric for both v0.5 and v1.0 dataset is AP across IoU threshold from 0.5 to 0.95. In LVIS v1.0, we train our model on 100k train images and valid it on 19k val set and also reported our final results on 19k testing set.

Dataset	**Model**	**EHLoss**	**SD**	AP	AP_r	AP_c	AP_f
LVIS v0.5	Cascade rcnn-X101 32x8d			28.24	18.31	27.25	33.44
LVIS v0.5	Cascade rcnn-X101 32x8d	V		30.42	18.65	31.44	33.82
LVIS v0.5	Cascade rcnn-X101 32x8d	V	V	30.86	20.37	31.3	34.48
LVIS v1.0	Cascade rcnn-X101 32x8d			28.15	15.9	27.36	34.62
LVIS v1.0	Cascade rcnn-X101 32x8d	V		29.1	18.02	28.12	34.6
LVIS v1.0	HTC-X101 32x8d			30.2	18.1	29.9	36.2
LVIS v1.0	HTC-X101 32x8d	V		30.9	18.67	30.0	37.5
LVIS v1.0	HTC-X101 32x8d	V	V	31.2	20.97	30.2	36.8

Table 1. The results of ablation on expert head loss (EHLoss) and synthetic data (SD) on LVIS val set. The metrics are mask AP and subscripts 'r', 'c', and 'f' stands for rare, common, and frequent category. Note that all model are trained using Repeat factor sampling (RFS) [9] with t = 0.001 and the HTC models are training with SyncBN [10] and Deformable Convolution [4] (DCN)

3.1 Multi-Expert Head Loss

In this experiment, we perform our method on LVIS v0.5 and v1.0 dataset. Considering the computation cost, we only do the experiments on cascade rcnn [2]. However, we later add the expert loss on the Hybrid Task Cascade (HTC) [3] model to improve the performance in the challenge. The result is shown in Table 1.

3.2 Synthetic Instance Augmentation

In this section, we apply our external synthetic instance augmentation on cascade rcnn [1] and HTC [3] model. The backbone of all cascade rcnn and HTC model are ResNeXt-X101-32x-8d and also equipped with FPN [8]. We randomly select 1000 images without annotations in LVISv1.0 dataset and random paste the

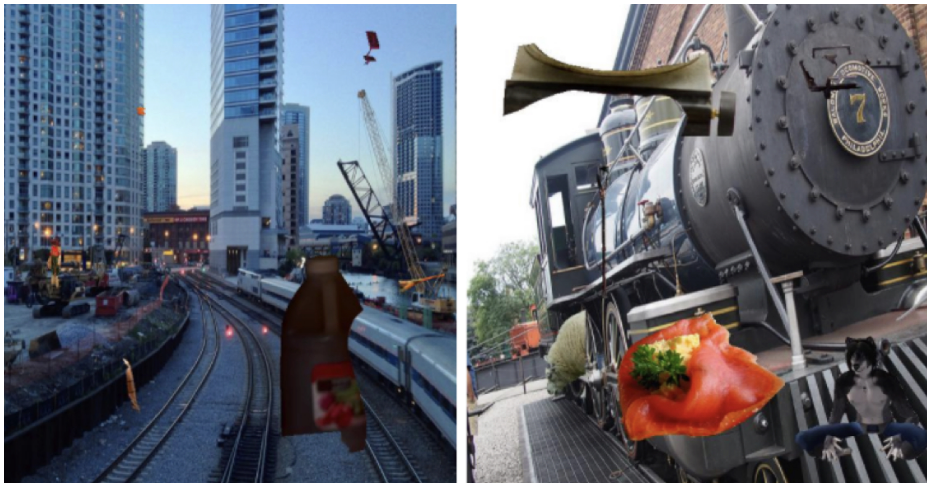

Fig. 1. Examples of the synthetic instance augmentation.

instance segmentations to create synthetic images. We paste the instances with the proportion rare: common: frequency = 1: 3: 5. The result is demonstrated in Table 1. Some examples for synthetic data are shown in Figure 1.

Model	AP	AP_r	AP_c	AP_f
Challenge baseline	30.2	18.1	29.9	36.2
+Expert Head	30.9	18.67	30.0	37.5
+Synthetic Data	31.2	20.97	30.2	36.8
+Two-phase Training	33.1	25.2	32.8	36.9
+Multi-scale Testing	34.1	24.3	33.92	38.82

Table 2. Experiment results of different tricks on LVIS val set.

4 LVIS Challenge 2020 and Ablation Study

We use several methods to enhance the model for the challenge. The gain by each term is shown in Table Table 2. With those enhancements, we achieve 34.1 and 33.7 Mask AP on val and test-dev set respectively. The comparison results between LVIS [5] are demonstrated in Table 3.

Challenge Baseline. In this challenge, we use Hybrid Task Cascade (HTC) [3] with ResNeXt101-64x-4d as backbone as our baseline model and use synchronized batch normalization [10] (SyncBN) in backbone and heads. Additionally, we apply deformable convolution [4] (DCN) in stage3, stage4 and stage5 of the model. In the training phase, We use multi-scale training and the shorter edge of images are resized to range from 400 to 1400, and the longer edge is set to

Model	eval. set	AP	AP_r	AP_c	AP_f
Official Baseline	val	27.2 ± 0.17	19.6 ± 0.50	26.0 ± 0.33	31.9 ± 0.06
Ours†	val	34.1 ± 0.14	24.3 ± 0.42	33.9 ± 0.28	38.8 ± 0.08
Official Baseline	test-dev	26.8	19.0	25.2	32.0
Ours	test-dev	33.7	23.9	33.5	38.4

† Due to computation cost, the results are average of the 3 runs.

Table 3. Final results on val and test set.

1400. We only apply horizontal flipping as our default data augmentation. Also, we use class-specific mask and box prediction to achieves better performance. In the testing phase, following the experiments in LVIS paper [5], we reduce the score threshold from 0.05 to 0.0001, and we select top 300 bounding boxes as the detection results. Other settings are kept the same as origin implementation if not mentioned.

Multi-scale Testing We apply multi-scale testing on both bounding box and segmentation results. The testing scales are set to (1333, 800), (1467,880) and (1600,960) randomly.

Synthetic Data and Repeat factor sampling. Due to the rules of the challenge, we are not permitted to use external data. However, if we only use those images without annotations in training data, the synthetic data we can make will be quite insufficient comparing with the entire training set. Therefore, we apply repeat factor sampling with t = 0.001 on training data and random sample 5000 images from the repeat sampling images set. We combine 5000 images from RFS and 1000 images from our synthetic instance set to get the external data for training.

Two-phase Training. To enhance our model, we apply the two-phase training method. Following the balanced group softmax [7], in the first stage, we reduce the effect from the repeat factor sampling which will damage the representation learning. In the second stage, we apply balanced group softmax on our fully connected layers of classifiers to learn the classification balancedly. This two-phase training method improve the AP by 1.9. The improvement is consistent with the experiment done in the paper, which means that our representation learning is better than using only repeat factor sampling.

5 Conclusion

We propose a expert head and synthetic instance augmentation method for improving the performance of current state-of-the-art model based on object detection and instance segmentation over long-tail dataset. Both of them are able to effectively improve the classification performance over the long-tail distribution data by enhancing the classification accuracy and reduce the destruction of representation learning. Currently, our strategy for Expert head and synthetic data augmentation is not optimized and we will investigate in future works.

References

1. Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6154–6162 (2018)
2. Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
3. Chen, K., Pang, J., Wang, J., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Shi, J., Ouyang, W., et al.: Hybrid task cascade for instance segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4974–4983 (2019)
4. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 764–773 (2017)
5. Gupta, A., Dollar, P., Girshick, R.: LVIS: A dataset for large vocabulary instance segmentation. In: CVPR (2019)
6. Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., Kalantidis, Y.: Decoupling representation and classifier for long-tailed recognition. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020 (2020), https://openreview.net/forum?id=r1gRTCVFvB
7. 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)
8. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
9. Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., van der Maaten, L.: Exploring the limits of weakly supervised pretraining. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 181–196 (2018)
10. Peng, C., Xiao, T., Li, Z., Jiang, Y., Zhang, X., Jia, K., Yu, G., Sun, J.: Megdet: A large mini-batch object detector. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6181–6189 (2018)
11. Zhou, B., Cui, Q., Wei, X.S., Chen, Z.M.: BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9719–9728 (2020)