FASA: Feature Augmentation and Sampling Adaptation for Long-Tailed Instance Segmentation

Yuhang Zang[1] Chen Huang[2] Chen Change Loy[1]✉
[1]S-Lab, Nanyang Technological University [2]Carnegie Mellon University
{zang0012, ccloy}@ntu.edu.sg chen-huang@apple.com

Appendix

In the supplementary materials, we discuss the ablation studies and implementation details that are not elaborated in the main paper. Section A highlights our FASA largely reduces the classification error. Section B analyzes some ablation studies of the adaptive feature sampling module. Section C validates the memory- and time-efficiency of FASA. Section D presents the clustering results used in the feature space. Section E provides the performance of FASA under the recently proposed AP^{Fixed} and AP^{Pool} metrics [8]. Section F reports our implementation details of feature augmentation methods. Section G compares FASA with another instance-level re-sampling based approach NMS re-sampling [20]. Section H shows the visualization result of FASA.

A. Error Analysis of Long-Tailed Instance Segmentation: Classification Error Dominates

In the main paper, we apply FASA only to the classification branch of Mask R-CNN. But why the choice? How does this simple mechanism impact performances of other branches like detection? To answer such questions, we need a detailed error analysis other than the default, single metric mean-average precision (mAP). We use the recent TIDE [1] toolbox to report six error metrics for long-tailed instance segmentation.

Table 1 outlines the six error metrics on large-scale LVIS dataset. We see that for the Mask R-CNN baseline [12], **classification error** is the main bottleneck for long-tailed instance segmentation when compared to other error types, *e.g.*, localization error. This explains our use of FASA to the classification branch of Mask R-CNN. Obviously, augmenting classification branch only will not incur a high cost. Performance-wise, we do see a significant reduction in classification error (from 25.10% to 20.74%) without deteriorating other errors much. With more budget, one could apply FASA to augment other branches of Mask R-CNN with the hope of more gains in all error metrics.

B. Ablation on Adaptive Feature Sampling

B.1. Initial Feature Sampling Probabilities

To initialize our adaptive virtual feature sampling process, we assigned class-wise sampling probabilities based on inverse class frequency. This scheme favors rare-class feature augmentation at the beginning, and does not rely on too many assumptions about skewed data distribution. Another sampling scheme that has minimal assumption of data distribution is based on uniform class distribution.

Table 2 compares the two schemes empirically. We see that both the uniform initialization and the inverse class frequency initialization boost the performance compared with the no augmentation baseline. Overall, the inverse class frequency initialization approach achieves better overall mask mAP. The AP_r and AP_f results of uniform initialization are slightly worse than initialization based on inverse class frequency. So we use the initialization based on inverse class frequency by default for its effectiveness and simplicity.

B.2. Performance Metric for Sampling Adaptation

In the main paper, we propose a virtual feature sampling approach that is adapted to the validation loss rather than validation metric. This is to avoid the large computational cost from frequent metric evaluation on large-scale dataset. Concretely, evaluating the validation metric of mAP on the dataset takes nearly 45 minutes, which is very expensive if we were to conduct it in each epoch. To test how much we can gain from adapting to the true performance metric, we compare the use of the two supervisory signals on a smaller task. We choose the long-tailed image classification task on CIFAR-100-LT [3], a much smaller dataset than LVIS. Evaluating per-class accuracy is as efficient as evaluating the loss on CIFAR-100-LT validation set. Table 3 shows a marginal improvement from using the performance metric, which when translated to large-scale dataset, may not be worth the large cost for metric evaluation.

Table 1: Error analysis of Mask R-CNN [12] with and without FASA. We use the TIDE [1] toolbox and report the six error types (%) on LVIS v1.0 *validation* set: classification error (E_{cls}), location error (E_{loc}), both classification and location error (E_{both}), duplicate detection error (E_{dupe}), background error E_{bkg} and missed ground truth error (E_{miss}). We observe the dominance of E_{cls} for Mask R-CNN, and hence apply FASA only to the classification branch of Mask R-CNN at minimum cost. This leads to big improvements in E_{cls} already. We expect more gains in E_{cls} and other error metrics by augmenting other branches of Mask R-CNN if given more computational budget.

Method	E_{cls}	E_{loc}	E_{both}	E_{dupe}	E_{bkg}	E_{miss}
Mask R-CNN	25.10	6.71	0.59	0.36	3.17	7.15
+ FASA	20.74	6.80	0.50	0.41	3.48	7.29
Δ	**-4.36**	+0.09	-0.09	+0.05	+0.29	+0.14

Table 2: Comparing different initialization schemes of our virtual feature sampling probabilities. AP, AP_r, AP_c and AP_f refer to the mask mAP metrics (%) for overall, rare, common and frequent class groups. The symbol 'FS' denotes to our adaptive Feature Sampling module.

FS	Initial sampling probability	AP	AP_r	AP_c	AP_f
✗	✗	22.3	12.7	21.7	26.9
✓	Uniform distribution	23.2	15.1	**23.4**	26.6
✓	Inverse class frequency	**23.7**	**17.8**	22.9	**27.2**

Table 3: Comparing the use of different performance metrics (validation loss vs. metric) for adaptive feature sampling on CIFAR-100-LT [3] dataset. The average and standard deviation of classification accuracy are from 3 runs.

Perf. measurement	Accuracy (%)
Validation loss	43.7 ± 0.25
Validation metric	$\mathbf{43.9 \pm 0.38}$

Table 4: Comparing the group-wise and class-wise feature sampling adaptation on CIFAR-100-LT [3] dataset. The average and standard deviation of classification accuracy are from 3 runs.

Sampling adaptation	Accuracy (%)
Group-wise	$\mathbf{43.7 \pm 0.25}$
Class-wise	43.3 ± 0.85

B.3. Group-wise vs. Class-wise Adaptation

By default, we adjust the feature sampling probability for each class group rather than each class. One of the reasons is that some classes may be missing for performance evaluation, *e.g.*, on LVIS validation set. This makes it impossible for loss-adapted sampling probability adjustment for every class. But what if all classes are available for evaluation, appearing on both training and validation sets?

We again test on the CIFAR-100-LT [3] dataset that meets the requirement. In this case, class grouping is not a *must* anymore, and our goal is to see if group-wise sampling adaptation still holds its benefits over class-wise sampling adaptation. Table 4 gives a positive answer. We observe that group-wise sampling adaptation performs better and has a lower variance since it relies on the stabler group-wise loss average rather than the noisy per-class loss.

Table 5: Comparison of training memory M_{train} and training time T_{train} required, with and without FASA on LVIS v1.0.

Method	FASA	M_{train} (GB)	T_{train} (s/iter)
Mask R-CNN + RFS	✗	11.1	0.768 ± 0.04
	✓	12.4	0.792 ± 0.05

C. Speed Analysis

Table 5 further validates the memory- and time-efficiency of our FASA approach. We see that FASA adds only a small amount of memory, which is used to maintain the online feature mean and variance of observed training samples. Thus the extra memory is constant and dependent only on the feature dimension. FASA is also found to incur a very small time cost.

D. Visualizing Class Grouping Results

Recall our feature sampling probabilities are adjusted in a group-wise manner. The class groups are formed by the Mean-shift [6] clustering algorithm. Figure 1 presents some clustering results, where visually similar or semantically related classes stay close in the feature space (*e.g.*, "hairnet" and "visor"). In addition, the co-occurrent classes also tend to stay close (*e.g.*, "pillow" and "loveseat"). Intuitively, related classes are better suited to have their sampling probabilities adjusted together.

E. Evaluation with AP^{Fixed} and AP^{Pool}

The mean average precision metric (denotes as AP^{Old} in the following) is the default evaluation metric for the instance segmentation task [16, 11]. Recently, Dave *et al.* [8] argued that the AP^{Old} metric is sensitive to changes in cross-category ranking, and introduced two complementary metrics AP^{Fixed} and AP^{Pool} to replace AP^{Old} for LVIS [11] dataset. Dave *et al.* [8] found that some methods improve AP^{Old} but have less impact on AP^{Fixed} and AP^{Pool}. The AP^{Old} metric limits the maximum detection results per image, resulting in cross-category competition. To address this issue, the AP^{Fixed} metric limits the maximum detection results per class on the dataset instead. To highlight the score calibration property, Dave *et al.* [8] also proposed AP^{Pool} metric that is class-agnostic and evaluates detection

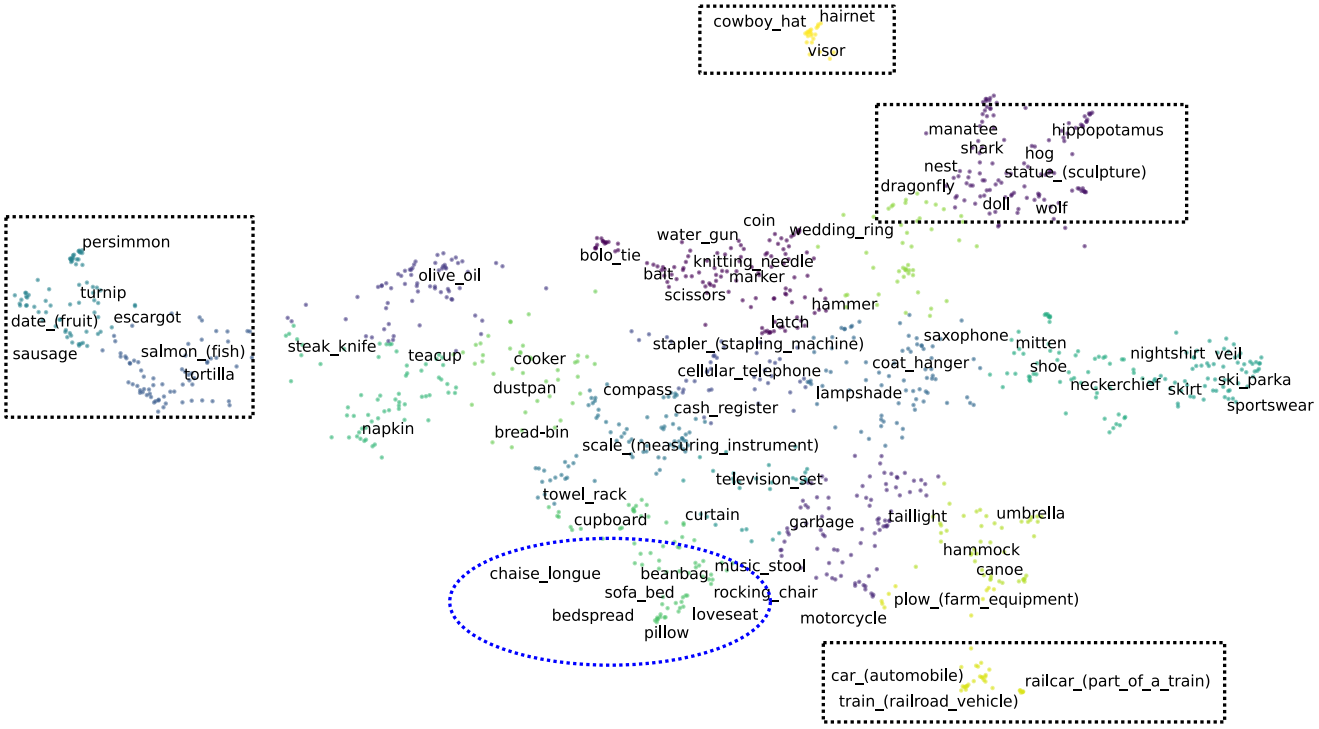

Figure 1: t-SNE [18] visualization of class groups. In the black dashed boxes, classes are often semantically related or visually similar. In the blue ellipse, we find classes that exhibit strong co-occurrence, *e.g.*, between 'pillow' and 'bedspread'.

results across all categories together. Since AP^{Pool} is class-agnostic, the evaluation is influenced more heavily by frequent classes rather than rare classes.

We provide the experimental results of FASA under the AP^{Fixed} and AP^{Pool} in Table 6. We observe that FASA consistently boosts the performance under AP^{Fixed} and AP^{Pool}, especially for the rare categories. For AP^{Fixed}, FASA improves overall AP and rare-class AP_r by 1.1% / 1.7% respectively for Mask R-CNN and 1.3% / 2.4% for Cascade Mask R-CNN. For AP^{Pool}, FASA still obtains 2.9% gains in AP_r for Mask R-CNN and 2.0% for Cascade Mask R-CNN. Since AP^{Pool} is mainly affected by the frequent class, the overall AP improvements are small. The AP^{Fixed} and AP^{Pool} results also demonstrate that FASA largely improves the performance of rare classes without compromising the common classes and frequent classes.

F. Implementation Details

Our implementation is based on the Mask R-CNN [12] backbone, which is the ImageNet pre-trained ResNet-50 [13] with a FPN [15] neck and a box head with two sibling fully connected layers for RoI classification and regression. We apply random horizontal image flipping and multi-scale jittering with the smaller image sizes (640, 672, 704, 736, 768, 800) in all experiments. All models are trained with standard SGD on 8 NVIDIA V100 GPUs. We follow the default settings in MMDetection [5] to set other hyper-parameters such as learning rates and training schedules.

Here we also describe the implementation details of the feature augmentation methods listed in Table 2 of the main paper. We detail the hyper-parameters of these approaches and our searched optimal choices in Table 7.

SMOTE [4]. The SMOTE algorithm interpolates neighboring features (*i.e.*, feature embeddings of region proposals) in the feature space. Specifically, for given features x_i, we consider the $k = 5$ nearest neighbours $\{x_j\}$ based on cosine feature distance. Then we interpolate new features as:

$$\hat{x} = \lambda \cdot x_i + (1 - \lambda) \cdot x_j, \tag{1}$$

where λ is a random value in $(0, 1]$.

MoEx [14]. Similar to SMOTE [4], the MoEx is also an interpolation-based augmentation method. As MoEx was developed for the image classification task, we transfer it to the instance segmentation task with the following modification: 1) we applied MoEx augmentation in the classifier of the Mask R-CNN [12] framework, 2) we searched the optimal value of parameters on the LVIS dataset and the results are shown in Table 7.

InstaBoost [10]. InstaBoost is an adaptive copy-and-paste FA method based on a location probability map. Since In-

Table 6: Results of Cascade Mask R-CNN [2] with and without FASA under the recently proposed AP^{Fixed} and AP^{Pool} metric [8]. We report the results on the LVIS [11] *validation* set. AP, AP_r, AP_c and AP_f refer to the mask mAP metrics (%) for overall, rare, common and frequent class groups. The symbol AP^{Old} refers to the standard mean average precision (mAP) metric. We observe FASA offers consistent performance boost under the AP^{Fixed} and AP^{Pool}, especially for the rare categories. All the models use the ResNet-101 [13] backbone and Repeat Factor Sampling (RFS) [11].

Method	FASA	AP^{Fixed}				AP^{Pool}			
		AP	AP_r	AP_c	AP_f	AP	AP_r	AP_c	AP_f
Mask R-CNN [12]	✗	27.1	20.3	26.9	30.3	27.2	9.0	22.5	27.5
	✓	**28.2** (+1.1)	**22.0** (+1.7)	**28.3**	**30.9**	**27.4** (+0.2)	**11.9** (+2.9)	**23.0**	**27.8**
Cascade Mask R-CNN [2]	✗	28.7	22.2	28.3	32.0	28.9	10.4	24.2	29.4
	✓	**30.0** (+1.3)	**24.6** (+2.4)	**29.8**	**32.4**	**29.2** (+0.3)	**12.4** (+2.0)	**25.0**	**29.6**

Table 7: Parameters tuned for the feature augmentation methods in Table 2 of the main paper.

Method	Param	Description	Value
MoEx, CVPR'21 [14]	p	MoEx probability	1.0
	ϵ	Epsilon constant for standard deviation	$1e^{-5}$
Liu *et al.*, CVPR'20 [17]	s	Scaling factor	20
	m_a	Angular margin	0.1
Chu *et al.*, ECCV'20 [7]	$\mathcal{T}_s$	Threshold to extract the class-specific features	0.3
	$\mathcal{T}_g$	Threshold to extract the class-generic features	0.6
Yin *et al.*, CVPR'19 [21]	α_{recon}	Coefficient of the reconstruction loss	0.5
	α_{reg}	Coefficient of the regularization loss	0.25

staBoost was already developed for the instance segmentation task, we use its default hyper-parameters.

Liu *et al.* [17]. Liu *et al.* [17] propose to transfer the angular distribution of face recognition loss such as CosFace [19] or ArcFace [9]. We select ArcFace for re-implementation:

$$L = -\frac{1}{N}\sum_{n=1}^{N}\log\frac{e^{s(\cos(\theta_y+\alpha_y+m_a))}}{e^{s(\cos(\theta_y+\alpha_y+m_a))}+\sum_{j\neq y}^{C}e^{s(\cos(\theta_j+\alpha_y))}} \tag{2}$$

The symbol θ_y refers to the angle between the input feature and the weight of the classifier. The symbol α_y means the extra angular that transfers from head class to tail class. As shown in Eq (2), two parameters are involved: the symbol s means the scaling factor applied to logit, and the symbol m_a refers to the angular margin. We tune these two parameters and show the results in Table 7.

We observed that since the instance segmentation task has to deal with the special *background* class, the margin-based ArcFace loss is unfortunately very sensitive to hyper-parameter choices of m_a. So margin-based augmentation [17] does not perform well on LVIS. Different from Liu *et al.* [17], our FASA is not limited to the form of loss functions.

Chu *et al.* [7]. Chu *et al.* [7] mixed the class-specific features of each class and the corresponding class-generic features of 'confusing' classes to synthesize new data samples. The definitions of 'class-generic' and 'class-specific' are based on the threshold masking of class activation map (CAM) [22]. For each real sample in the tail class, the authors sample N_a images from its N_f confusing classes.

During the re-implementation, we found that the difference in batch size between the classification and instance segmentation tasks limited the performance of Chu *et al.* [7] when transferred to the instance segmentation task. The classification task has a large batch size (*e.g.*, 128) that can meet the demand of picking confusing categories (*e.g.*, $N_a = N_f = 3$). However, instance segmentation models are limited by small batch size (*e.g.*, 2) and there is no guarantee that the top confusing categories will appear in the same batch. Compared with Chu *et al.* [7], our FASA builds feature banks for each category to cache the features of the previous batch, thus getting rid of the small batch limitation.

Yin *et al.* [21]. Yin *et al.* [21] is a feature augmentation method designed for the face recognition task. A total of three loss functions are included: face classification loss $\mathcal{L}_{sfmx}$, reconstruction loss $\mathcal{L}_{\text{recon}}$ and regularization loss $\mathcal{L}_{\text{reg}}$. The reconstruction loss $\mathcal{L}_{\text{recon}}$ is critical to train the discriminative feature encoder and decoder. To transfer into the instance segmentation task, we apply the reconstruction loss to the feature embedding of each positive region proposal. Besides, Yin *et al.* [21] need a two-stage training pipeline. In the first stage, the authors fix the backbone and generate new feature samples to train the classifier. In the second stage, the authors fix the classifier and update the other components. Such a two-stage approach introduces

Table 8: Comparing our FASA with NMS Re-sampling [20] on LVIS v1.0 *validation* set. The symbol 'NR' denotes the NMS Re-sampling approach. AP, AP_r, AP_c and AP_f refer to the mask mAP metrics (%) for overall, rare, common and frequent class groups.

NR	FASA	AP	AP_r	AP_c	AP_f
✗	✗	19.3	1.2	17.4	**29.3**
✗	✓	22.6	10.2	21.6	29.2
✓	✗	21.7	8.6	20.4	29.0
✓	✓	**22.9**	**11.1**	**21.8**	29.2

additional training time cost. Compared to Yin *et al*. [21], our FASA leverages end-to-end training and therefore more efficient.

G. Comparison with NMS Re-sampling [20]

NMS Re-sampling [20] is proposed to adjusting the NMS threshold for different categories during the training. Specifically, the NMS thresholds for the frequent/common/rare categories are set as $\{0.7, 0.8, 0.9\}$ with the increasing trend. Such a mechanism is beneficial to preserve more region proposals from the rare classes and suppress the number of proposals from frequent classes.

We compare the FASA with NMS Re-sampling in Table 8. The first line refers to the Mask R-CNN [12] baseline without any re-sampling or augmentation method. From the second and the third line, we see that both FASA and NMS Re-sampling achieve better performance than the baseline method. FASA performs slightly better than NMS Re-sampling, especially for the rare classes. We believe such a performance gap is due to NMS Re-Sampling is mainly in adjusting the sampling weights of the current data samples, while FASA can further generate *new* virtual samplers. Also, the bottom results demonstrate that FASA as an orthogonal module can combine with NMS Re-sampling to further boost the performance.

H. Result Visualization

To better interpret the result, we show the segmentation results of the selected rare classes in Figure 2. We observe that without FASA, the prediction scores for rare classes are small or even missed. On the contrary, with the help of our FASA, the classification results of the rare classes become accurate.

References

[1] Daniel Bolya, Sean Foley, James Hays, and Judy Hoffman. TIDE: A general toolbox for identifying object detection errors. In *ECCV*, 2020. 1, 2

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: High quality object detection and instance segmentation. *TPAMI*, 2019. 4

[3] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *NIPS*, 2019. 1, 2

[4] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. SMOTE: Synthetic minority oversampling technique. *JAIR*, 16:321–357, 2002. 3

[5] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open MMLab detection toolbox and benchmark. *arXiv preprint:1906.07155*, 2019. 3

[6] Yizong Cheng. Mean shift, mode seeking, and clustering. *TPAMI*, 17(8):790–799, 1995. 2

[7] Peng Chu, Xiao Bian, Shaopeng Liu, and Haibin Ling. Feature space augmentation for long-tailed data. In *ECCV*, 2020. 4

[8] Achal Dave, Piotr Dollár, Deva Ramanan, Alexander Kirillov, and Ross Girshick. Evaluating large-vocabulary object detectors: The devil is in the details. *arXiv preprint arXiv:2102.01066*, 2021. 1, 2, 4

[9] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *CVPR*, pages 4690–4699, 2019. 4

[10] Hao-Shu Fang, Jianhua Sun, Runzhong Wang, Minghao Gou, Yong-Lu Li, and Cewu Lu. Instaboost: Boosting instance segmentation via probability map guided copy-pasting. In *CVPR*, pages 682–691, 2019. 3

[11] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 2, 4

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 1, 2, 3, 4, 5

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 4

[14] Boyi Li, Felix Wu, Ser-Nam Lim, Serge Belongie, and Kilian Q Weinberger. On feature normalization and data augmentation. In *CVPR*, 2021. 3, 4

[15] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 3

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 2

[17] Jialun Liu, Yifan Sun, Chuchu Han, Zhaopeng Dou, and Wenhui Li. Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In *CVPR*, pages 2970–2979, 2020. 4

[18] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *JMLR*, 9(Nov):2579–2605, 2008. 3

[19] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. CosFace: Large margin cosine loss for deep face recognition. In *CVPR*, pages 5265–5274, 2018. 4

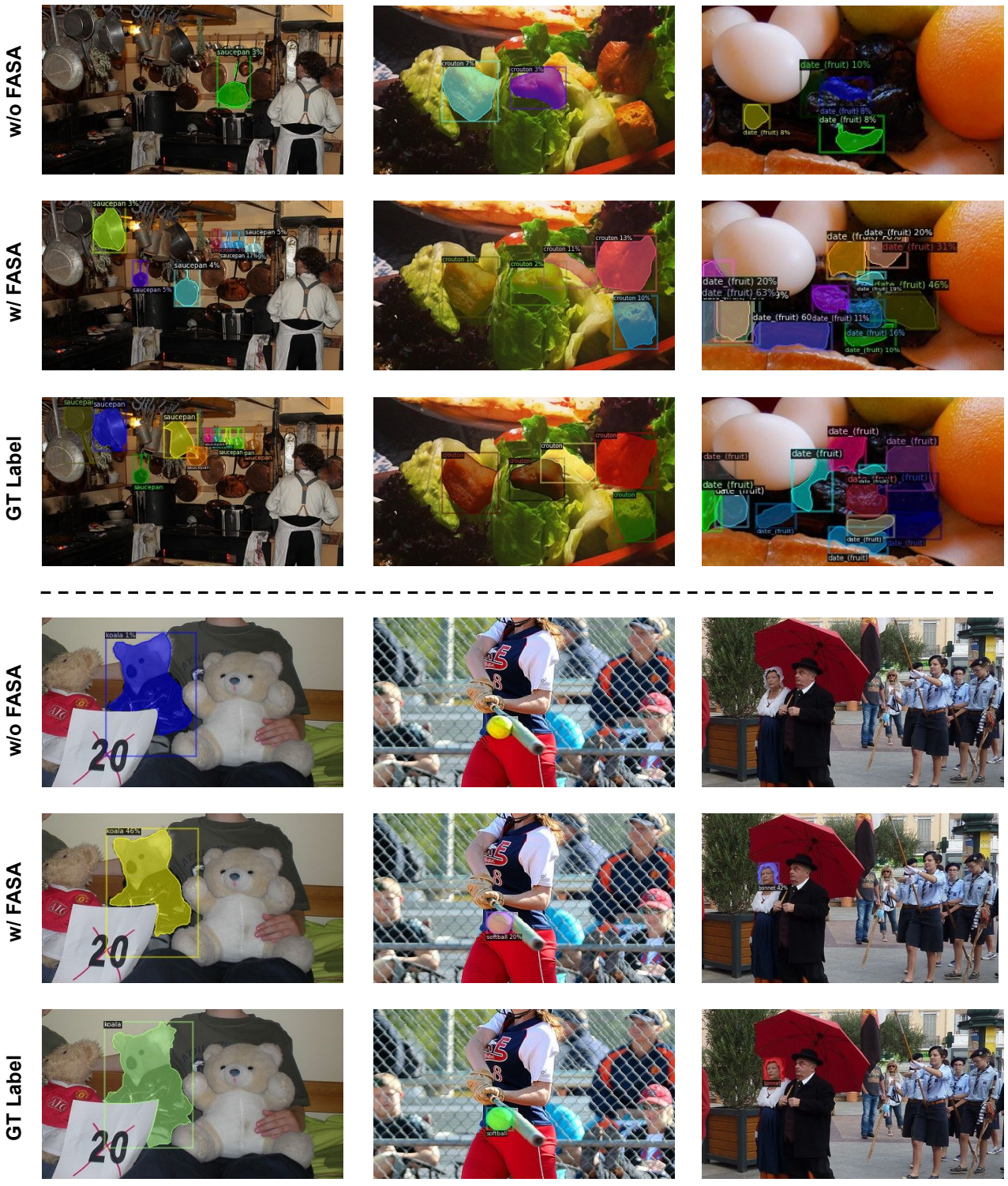

Figure 2: Prediction results of Mask R-CNN framework without and with FASA on the LVIS v1.0 *validation* set. We select six rare classes 'saucepan', 'crouton', 'date (fruit)', 'koala', 'softball' and 'bonnet' to visualize. We observe that with the help of FASA, Mask R-CNN exhibits more correct classification results than the baseline.

[20] Jialian Wu, Liangchen Song, Tiancai Wang, Qian Zhang, and Junsong Yuan. Forest R-CNN: Large-vocabulary long-tailed object detection and instance segmentation. In *ACM MM*, pages 1570–1578, 2020. 1, 5

[21] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. Feature transfer learning for face recognition with under-represented data. In *CVPR*, 2019. 4, 5

[22] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 4