On Calibration of Long-Tailed Object Detectors

Anonymous Author(s)
Affiliation
Address
email

Abstract

Vanilla models for object detection and instance segmentation suffer from the heavy bias toward detecting frequent objects in the long-tailed setting. Existing methods address this issue mostly during training, *e.g.*, by re-sampling or re-weighting. In this paper, we investigate a largely overlooked approach — *post-processing calibration* of confidence scores. We propose NorCal, **Normalized Calibration** for long-tailed object detection and instance segmentation, a simple and straightforward recipe that reweighs the predicted scores of each class by its training size. We show that separately handling the background class and normalizing the scores over classes for each proposal are keys to achieving superior performance. On the LVIS dataset, NorCal in diverse scenarios effectively improves nearly all baselines not only on rare classes but also on common and frequent classes. Finally, we conduct extensive analysis and ablation studies to offer insights into various modeling choices and mechanisms of our approach.

1 Introduction

Object detection and instance segmentation are the fundamental tasks in computer vision and have been approached from various perspectives over the past few decades [7, 12, 24, 33, 48]. With the recent advances in neural networks [1, 4, 9, 16, 27, 28, 30, 34, 37, 38, 40], we have witnessed an unprecedented breakthrough in detecting and segmenting frequently seen objects such as people, cars, and TVs [13, 14, 19, 29, 64]. Yet, when it comes to detect rare, less commonly seen objects (*e.g.*, walruses, pitchforks, seaplanes, etc.) [11, 47], there is a drastic performance drop largely due to insufficient training samples [43, 62]. How to overcome the "long-tailed" distribution of different object classes [63] has therefore attracted increasing attention lately [25, 42, 50].

To date, most existing works tackle this problem in the *model training phase*, *e.g.*, by developing algorithms, objectives, or model architectures to balance among the classes [11, 18, 25, 42, 44, 49, 50, 52, 53]. Wang et al. [50] investigated the widely used instance segmentation model, Mask R-CNN [16], and found that the performance drop comes primarily from *mis-classification of object proposals*. Concretely, the model tends to give frequent classes higher confidence scores [6], hence biasing the label assignment towards frequent classes. This observation suggests that techniques of class-imbalanced learning [2, 5, 15, 39] can be applied to long-tailed detection and segmentation.

Building upon the aforementioned observation, we take another route in the *model inference phase* by explicitly *post-processing calibration* [2, 20, 21, 31, 55], which adjusts a classifier's confidence scores among classes, without changing its internal weights or architectures. Post-processing calibration is efficient and widely applicable since it requires no re-training of the classifier. Its effectiveness on multiple imbalanced classification benchmarks [20, 54] may also translate to long-tailed object detection and instance segmentation.

In this paper, we propose a simple post-processing calibration technique inspired by imbalanced learning [31, 55] and show it can significantly improve a pre-trained object detector's performance

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

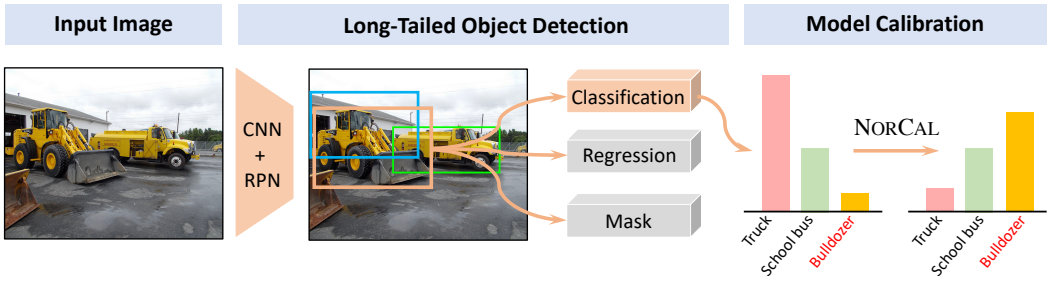

Figure 1: **Normalized Calibration** (NorCal)**.** Object detection or instance segmentation models trained with data from a long-tailed distribution tend to output higher confidence scores for the head classes (*e.g.*, "Truck") than for the tail ones (*e.g.*, the true class label "Bulldozer"). NorCal investigates a simple but largely overlooked approach to correct this mistake — model calibration of the classification scores *post-training* — and significantly improves nearly all the models we consider.

on detecting both rare and common classes of objects. *Note that our results are in sharp contrast to a couple of previous attempts on exploring calibration in object detection [6, 25], which reported poor performance and/or sensitivity to hyper-parameter tuning.*

Concretely, we apply post-processing calibration to the classification sub-network of a pre-trained object detector. Taking Faster R-CNN [40] and Mask R-CNN [16] for examples, they apply to each object proposal a $(C+1)$-way softmax classifier, where C is the number of foreground classes, and 1 is the background class. To prevent the scores biasing toward frequent classes [6, 50], we *re-scale the logit* of every class according to its class size, *e.g.*, number of training images. Importantly, we leave the logit of the background class intact because (a) the background class has a drastically different meaning from object classes and (b) its value does not affect the ranking among different foreground classes. After adjusting the logits, we then re-compute the confidence scores (with normalization across all classes, including the background) to decide the label assignment for each object proposal[1] (see Figure 1). We note that, it is crucial to normalize the scores across all classes since it triggers *re-ranking of the detection results within each class* (see Figure 3), influencing the class-wise precision and recall. Instead of separately adjusting each class by a specific factor [6], we follow [5, 31, 55] to set the factor as a function of the class size, leaving only one hyper-parameter to tune. We find that it is robust to use the training set to set this hyper-parameter, making our approach applicable to scenarios where collecting a held-out representative validation set is challenging.

Our approach, named **Normalized Calibration** for long-tailed object detection and instance segmentation (NorCal), is model-agnostic as long as the detector has a softmax classifier or multiple binary sigmoid classifiers for the objects and the background. We validate NorCal on the LVIS [11] dataset for both long-tailed object detection and instance segmentation. NorCal can consistently improve not only baseline models (*e.g.*, Faster R-CNN [40] or Mask R-CNN [16]) but also many models that are dedicated to the long-tailed distribution. Hence, our best results notably advance the state of the art. Moreover, NorCal can improve both the standard average precision (AP) and the category-independent AP^{Fixed} metric [6], implying that NorCal does not trade frequent class predictions for rare classes but rather *improve the proposal ranking within each class*. Indeed, through a detailed analysis, we show that NorCal can in general improve both the precision and recall for each class, making it appealing to almost any existing evaluation metrics. Overall, we view NorCal a simple plug-and-play component to improve object detectors' performance during inference.

2 Related Work

Long-tailed detection and segmentation. Existing works on long-tailed object detection can be categorized into re-sampling, cost-sensitive learning, and data augmentation. Among them, *Re-sampling methods* change the long-tailed training distribution into a more balanced one by sampling data from rare classes more often [3, 11, 41]. *Cost-sensitive learning* aims at adjusting the loss of data instances according to their labels [17, 42, 44, 49]. Building upon these, Some methods perform two-staged training, which first pre-train the model on balanced datasets [26] and fine-tune them on the target dataset using either re-sampling or cost-sensitive learning [18, 20, 25, 39, 50–

[1] Popular evaluation protocols allow multiple labels per proposal if their confidence scores are high enough.

52, 61]. Besides, another thread of works leverage data augmentation for the object instances in the tail classes to improve long-tailed object detection Ghiasi et al. [8], Ramanathan et al. [36], Zang et al. [59], Zhang et al. [60]. In contrast to all previous works, we investigate post-processing calibration [20, 21, 31, 46, 55] to adjust the learned model in the testing phase, without modifying the learning or modeling. Specifically, these methods adjust the predicted confident scores (*i.e.*, the posterior over classes) for each test instance, *e.g.*, by normalizing the classifier norms [20] or by scaling or reducing the logits with class sizes [21, 31, 55]. To our knowledge, only Li et al. [25] and Tang et al. [45] have studied this approach for object detection. Li et al. [25] applied classifier normalization [20] as a baseline but showed inferior results; Tang et al. [45] developed causal inference rules, which however require a corresponding de-confounded training step. Dave et al. [6] applied methods for calibrating model uncertainty, quite different from imbalanced learning (see Section 2 for a comparison). In this paper, we demonstrate that existing calibration rules for imbalanced learning [21, 31, 55] can significantly improve long-tailed object detection, even for a normally trained detector. We refer the reader to the supplementary materials for a comprehensive survey and comparison of the literature.

Calibration of model uncertainty. The calibration techniques we employ are different from the ones used for calibrating model uncertainty [10, 22, 23, 32, 35, 57, 58]: we aim to adjust the prediction across classes, while the latter adjusts the predicted probability to reflect the true correctness likelihood. Specifically for long-tailed object detection, Dave et al. [6] applied techniques for calibrating model uncertainty to each object class individually. Namely, a temperature factor or set of binning grids (*i.e.*, hyper-parameters) has to be estimated for each of the hundreds of classes in the LVIS dataset, leaving the techniques sensitive to hyper-parameter tuning. Indeed, Dave et al. [6] showed that it is quite challenging to estimate those hyper-parameters for tail classes. In contrast, the techniques we apply have only a single hyper-parameter, which can be selected robustly from the training data.

3 Post-Processing Calibration for Long-Tailed Object Detection

In this section, we provide the background and notation for long-tailed object detection and instance segmentation, describe our approach, **Normalized Calibration** for Long-tailed Object Detection and Instance Segmentation (NORCAL) and its properties, and discuss its relation to existing post-processing calibration methods.

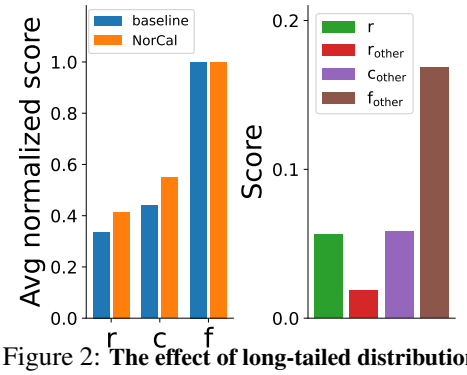

Figure 2: **The effect of long-tailed distribution on object detection. Left:** The scores from the baseline detector are biased to the frequent objects, which can be alleviated by NORCAL. **Right:** For rare objects, we further show their scores in **green** and the highest scores from another **rare**, **common**, and **frequent** classes on the same proposals. The long-tailed object detector is highly imbalanced towards frequent classes.

3.1 Background and Notation

Our tasks of interests are object detection and instance segmentation. Object detection focuses on detecting objects via bounding boxes while instance segmentation additionally requires precisely segmenting each object instance in an image. Both tasks involve classifying the object in each box/mask region into one of the pre-defined classes. This classification component is what our proposed approach focuses on improving. The most common object classification loss is the cross-entropy (CE) loss,

$$\mathcal{L}_{\text{CE}}(\boldsymbol{x}, \boldsymbol{y}) = -\sum_{c=1}^{C+1} y[c] \times \log\big(p(c|\boldsymbol{x})\big), \tag{1}$$

where $\boldsymbol{y} \in \{0,1\}^{C+1}$ is the ground-truth class and $p(c|\boldsymbol{x})$ is the probability (*i.e.*, confidence score) of the proposal $\boldsymbol{x}$ belonging to the class c, which is of the form

$$p(c|\boldsymbol{x}) = \frac{\exp(\phi_c(\boldsymbol{x}))}{\sum_{c'=1}^{C} \exp(\phi_{c'}(\boldsymbol{x})) + \exp(\phi_{C+1}(\boldsymbol{x}))}. \tag{2}$$

Here, ϕ_c is the logit for class c, which is usually realized by $\boldsymbol{w}_c^\top f_{\boldsymbol{\theta}}(\boldsymbol{x})$: $\boldsymbol{w}_c$ is the linear classifier associated with class c and $f_{\boldsymbol{\theta}}$ is the feature network. We use $C+1$ to denote the "background" class.

During testing, a set of (box/mask proposal, object class, confidence score) tuples are proposed for each image. The most common evaluation metric for these tuples is average precision (AP), where they are compared against the ground-truths *for each class*[2]. Concretely, the tuples with predicted class c will be gathered, sorted by their scores, and compared with the ground-truths for class c. Further, for popular benchmarks such as MSCOCO [26] and LVIS [11], there is a cap K (often set to 300) on the number of detected objects per image, which is enforced usually by including only the tuples with top K confidence scores.

Long-tailed object detection and instance segmentation: Problem and Empirical Evidence. Let N_c denote the number of training images of class c. A major challenge in long-tailed object detection is that N_c is imbalanced across classes, and the learned classifier using Eq. 1 is biased toward giving higher scores to the head classes (whose N_c is larger) [2, 5, 6, 15]. For instance, in the long-tailed object detection benchmark LVIS [11] whose classes are divided into frequent ($N_c > 100$), common ($100 \geq N_c > 10$), and rare ($N_c \leq 10$), the top K tuples (during inference) often belong to the frequent classes. In Figure 2, we show that the confidential scores of rare classes are much smaller than the frequent classes, making each proposal on rare objects difficult to compete with the objects from frequent classes, which aligns with Wang et al. [50] that rare objects are often mis-classified.

Recently, Dave et al. [6] point out that the standard AP metric can be gamed in long-tailed object detection and instance segmentation, so they propose several new metrics with modified limits on the number of detected objects, per image or across the entire test set. This issue is inherently related to inaccurate confidence scores in the long-tailed setting. In our experiments, we show that our approach works robustly well with their proposed change to the evaluation metric.

3.2 Normalized Calibration for Long-tailed Object Detection (NorCal)

Post-Processing Calibration and Foreground-Background Decomposition. We explore applying simple post-calibration techniques from standard multi-way classification [20, 54] to object detection and instance segmentation. The main idea is to scale down the logit of each class c by its size N_c [31, 55]. In our case, however, the background class posts a unique challenge. First, N_{C+1} is ill-defined since nearly all images contain backgrounds. Second, the background patches extracted during model training are drastically different from the foreground object proposals in terms of amounts and appearances. We thus propose to decompose Eq. 2 as follows,

$$p(c|\boldsymbol{x}) = \frac{\sum_{c'=1}^{C} \exp(\phi_{c'}(\boldsymbol{x}))}{\sum_{c'=1}^{C} \exp(\phi_{c'}(\boldsymbol{x})) + \exp(\phi_{C+1}(\boldsymbol{x}))} \times \frac{\exp(\phi_c(\boldsymbol{x}))}{\sum_{c'=1}^{C} \exp(\phi_{c'}(\boldsymbol{x}))}, \tag{3}$$

where the first term on the right-hand side predicts how likely $\boldsymbol{x}$ is foreground (vs. background, *i.e.*, class $C+1$) and the second term predicts how likely $\boldsymbol{x}$ belongs to class c given that it is foreground. Note that, the background logit $\phi_{C+1}(\boldsymbol{x})$ only appears in the first term and is compared to all the foreground classes as a whole. In other words, scaling or reducing it does not change the order of confidence scores among the object classes $c \in \{1, \cdots, C\}$. We thus choose to keep $\phi_{C+1}(\boldsymbol{x})$ intact. Please refer to Section 4 for a detailed analysis, including the performance of the foreground vs. background classifier and the effect of adjusting $\phi_{C+1}(\boldsymbol{x})$.

For the foreground object classes, inspired by Figure 2 and the studies in [6, 31, 55], we propose to reduce the exponential of the logit $\phi_c(\boldsymbol{x}), \forall c \in \{1, \cdots, C\}$, by a positive factor a_c,

$$p(c|\boldsymbol{x}) = \frac{\exp(\phi_c(\boldsymbol{x}))/a_c}{\sum_{c'=1}^{C} \exp(\phi_{c'}(\boldsymbol{x}))/a_{c'} + \exp(\phi_{C+1}(\boldsymbol{x}))}, \tag{4}$$

in which a_c should monotonically increase with respect to N_c — such that the scores for head classes will be suppressed. We investigate a simple way to set a_c, inspired by [21, 55],

$$a_c = N_c^{\gamma}, \;\; \gamma \geq 0, \tag{5}$$

which has a single hyper-parameter γ that controls the strength of dependency between a_c and N_c. Specifically, if $\gamma = 0$, we recover the original confidence scores in Eq. 2. We investigate other methods beyond Eq. 4 and Eq. 5 in Section 4.

[2]The difference between APs for object detection and instance segmentation lies in the computation of the intersection over union (IoU)

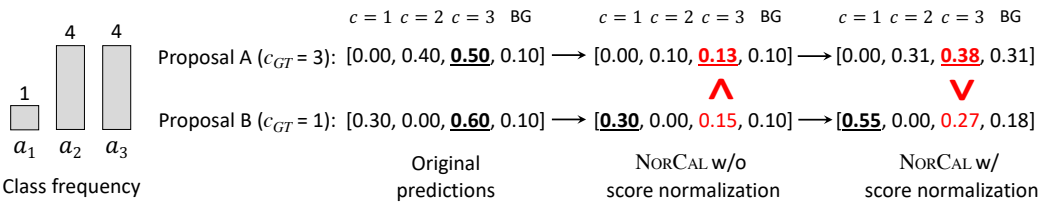

Figure 3: NorCal **with score normalization can improve AP for head classes.** The proposed calibration approach and score normalization can be organically coupled together to improve long-tailed object detection. Bold and underlined number indicates the largest class for each proposal. See the text for details.

Hyper-parameter. Our approach only has a single hyper-parameter γ to tune. We observe that we can tune γ directly on the training data[3], bypassing the need to collect a held-out set, which can be hard to construct due to the scarcity of examples for the tail classes. Dave et al. [6] also investigate this idea; however, the selected hyper-parameters from training data hurt the test results of rare classes. We attribute this to the fact that their methods have separate hyper-parameters for each class, and that makes them hard to tune.

The importance of normalization and its effect on AP. At first glance, our approach NorCal seems to simply scale down the scores for head classes, and may unavoidably hurt their AP due to the decrease of detected tuples (hence the recall) within the cap. However, we point out that the normalization operation (*i.e.*, sum to 1) in Eq. 4 can indeed improve AP for head classes — *normalization enables re-ordering the scores of tuples within each class.*

Let us consider a three-class example (see Figure 3), in which $c = 1$ is a tail class, $c = 2$ and $c = 3$ are head classes, and $c = 4$ is the background class. Suppose two proposals are found from an image: proposal A has scores $[0.0, 0.4, 0.5, 0.1]$ and the true label $c = 3$; proposal B has scores $[0.3, 0.0, 0.6, 0.1]$ and the true label $c = 1$. Before calibration, proposal B is ranked higher than A for $c = 3$, resulting in a low AP. Let us assume $a_1 = 1$ and $a_2 = a_3 = 4$. If we simply divide the scores of object classes by these factors, proposal B will still be ranked higher than A for $c = 3$. However, by applying Eq. 4, we get the new scores for proposal A as $[0.0, 0.31, 0.38, 0.31]$ and for proposal B as $[0.55, 0.0, 0.27, 0.18]$ — proposal A is now ranked higher than B for $c = 3$, leading to a higher AP for this class. As will be seen in Section 4, such a "re-ranking" property is the key to making NorCal excel in AP for all classes as well as in other metrics like AP^{Fixed} [6].

3.3 Comparison to Existing Work

Li et al. [25] investigate classifier normalization [20] for post-processing calibration. Concretely, they modify the calculation of ϕ_c from $\boldsymbol{w}_c^\top f_{\boldsymbol{\theta}}(\boldsymbol{x})$ to $\frac{\boldsymbol{w}_c^\top}{\|\boldsymbol{w}_c\|_2^\gamma} f_{\boldsymbol{\theta}}(\boldsymbol{x})$, building upon the observation that the classifier weights of head classes tend to exhibit larger norms. The results, however, are much worse than their proposed method BaGS. They attribute this to their handling of the background class, where they combine two models, with or without classifier normalization, to improve the baseline. Our decomposition in Eq. 3 suggests a more straightforward way to handle the background class. Moreover, N_c provides a better signal for calibration than $\|\boldsymbol{w}_c\|_2$, according to [21, 55].

3.4 Extension to Multiple Binary Sigmoid Classifiers

Many existing models for long-tailed object detection and instance segmentation are based on multiple binary classifiers instead of the softmax classifier [17, 39, 42, 44, 51]. That is, s_c in Eq. 2 becomes

$$s_c = \frac{1}{1 + \exp(-\phi_c(\boldsymbol{x}))} = \frac{1}{1 + \exp(-\boldsymbol{w}_c^\top f_{\boldsymbol{\theta}}(\boldsymbol{x}))}, \tag{6}$$

in which $\boldsymbol{w}_c$ treats every class $c' \neq c$ and the background class as the "negative" class. In other words, the background logit $\phi_{C+1} = \boldsymbol{w}_{C+1}^\top f_{\boldsymbol{\theta}}(\boldsymbol{x})$ in Eq. 2 is not learned.

[3] Unlike imbalanced classification in which the learned classifier ultimately achieves $\sim 100\%$ accuracy on the training data [55, 56] (so hyper-parameter tuning using the training data becomes infeasible), a long-tailed object detector can hardly achieve 100% AP per class even on the training data.

Our post-processing calibration approach can be extended to multiple binary classifiers as well. For example, Eq. 4 becomes

$$s_c = \frac{1}{1 + \exp(-\phi_c(\boldsymbol{x})) \times a_c}. \tag{7}$$

However, one drawback of this (which we will empirically demonstrate in the experiments) is the infeasibility of normalization across classes; s_c does not necessarily sum to 1. Forcing them to sum to 1 would inevitably turn many background patches into foreground proposals due to the lack of the background logit ϕ_{C+1}.

4 Experiments

4.1 Setup

Dataset. We benchmark NorCal on the LVIS v1 dataset [11], which has 100K/19.8K/19.8K training/validation/test images, a standard benchmark dataset for large-vocabulary instance segmentation. There are 1203 categories, divided into three groups based on the number of training images per class: rare (1–10 images), common (11–100 images), and frequent (>100 images). All results are reported on the validation set. Our main focus is instance segmentation on v1. For comparisons to existing work, we also conduct experiments and analyses on LVIS v0.5 (see the appendix for more details).

Evaluation metrics. We adopt the standard mean Average Precision (AP) [26] for evaluation. Following [11], we denote the AP for rare, common, and frequent categories as AP_r, AP_c, and AP_f, respectively. We also report results with a complementary metric AP^{Fixed} [6], which replaces the cap over detections-per-image with a per-class cap over the entire evaluation set. Meanwhile, $\text{AP}^b$is also reported for instance segmentation results, which assess the AP for the bounding boxes induced by the instance segmentation models.

Implementation details and variants of NorCal**.** For the baseline detectors, we use the released checkpoints from corresponding papers. We report the results on the validation set with the best hyper-parameter tuned on training images. For NorCal, (a) we investigate different mechanisms by applying post-calibration on the classifier logits, exponentials, or probabilities; (b) we study different types of calibration factor a_c, using class-dependent temperature (CDT) [55] or effective number of samples (ENS) [5], and (c) we compare with or without score normalization.

4.2 Main results

NorCal **effectively improves baselines in diverse scenarios.** We first apply NorCal on top of multiple representative baselines for instance segmentation: (1) Mask R-CNN [16] with feature pyramid networks [27], which is trained with repeated factor sampling, following the standard training procedure in [11]. (2) Re-weighting based methods that have a multi-class classifier, *e.g.*, cRT [20] and LWS [20]; (3) Re-weighting based methods that trained with multiple binary classifiers, *e.g.*, EQL [42]; (4) Data augmentation based methods, *e.g.*, MosaicOS [60]. All models are adopted from the original papers. *Please see appendix for a comparison with more existing methods.*

Table 1 provides our main results on LVIS v1. NorCal achieves consistent gains on top of all the models of different backbone architectures. For instance, with ResNet-50 and RFS [11], the overall AP improves from 22.58 to 25.22, including about 7 AP_r and 3 AP_c gains on rare and common objects, respectively. Furthermore, NorCal improves upon the methods that are specifically designed for long-tailed object detection and instance segmentation: LWS [20] cRT [20], and MosaicOS [60].

Moreover, we note that NorCal's benefit is captured by almost all evaluation metrics (columns) except a slight performance drop on frequent classes when applying NorCal to EQL, which is trained with multiple binary classifiers. We attribute this to the infeasibility of normalization across classes (Section 3.4).

These results demonstrate a key strength of NorCal that is not commonly observed in the literature: the overall gain without sacrificing much of the accuracy on frequent classes. We attribute this to the score normalization operation of NorCal; unlike existing post-processing calibration methods [6,

Table 1: **Comparison of instance segmentation on the validation set of LVIS v1.** NorCal provides solid improvement to existing models to which it is applicable. †: with EQL, we see slight performance drop on the frequent class due the infeasibility of normalization across classes with multiple binary classifiers.

Backbone	Method	NorCal	AP	AP_r	AP_c	AP_f	AP^b
R-50	EQL [42]	✗	18.60	2.10	17.40	27.20	19.30
		✓	(+2.30) 20.90	(+3.90) 6.00	(+3.80) 21.20	†(-0.10) 27.10	(+2.50) 21.80
	LWS [20]	✗	17.00	2.00	13.50	27.40	17.50
		✓	(+3.00) 20.00	(+2.90) 4.90	(+5.30) 18.80	(+0.40) 27.80	(+3.20) 20.70
	cRT [20]	✗	22.10	11.90	20.20	29.00	22.20
		✓	(+2.20) 24.30	(+3.50) 15.40	(+2.70) 22.90	(+0.70) 29.70	(+1.50) 23.70
	RFS [11]	✗	22.58	12.30	21.28	28.55	23.25
		✓	(+2.65) 25.22	(+7.03) 19.33	(+2.88) 24.16	(+0.43) 28.98	(+2.83) 26.08
	MosaicOS [60]	✗	24.45	18.17	22.99	28.83	25.05
		✓	(+2.32) 26.76	(+5.69) 23.86	(+2.82) 25.82	(+0.27) 29.10	(+2.73) 27.77
R-101	RFS [11]	✗	24.82	15.18	23.71	30.31	25.45
		✓	(+2.43) 27.25	(+5.61) 20.79	(+2.74) 26.45	(+0.68) 30.99	(+2.60) 28.05
	MosaicOS [60]	✗	26.73	20.53	25.78	30.53	27.41
		✓	(+2.30) 29.03	(+5.85) 26.38	(+2.37) 28.15	(+0.66) 31.19	(+2.55) 29.96
X-101	RFS [11]	✗	26.67	17.60	25.58	31.89	27.35
		✓	(+1.25) 27.92	(+2.15) 19.75	(+1.61) 27.19	(+0.45) 32.34	(+1.49) 28.83
	MosaicOS [60]	✗	28.29	21.75	27.22	32.35	28.85
		✓	(+1.52) 29.81	(+3.97) 25.72	(+1.70) 28.92	(+0.24) 32.59	(+1.71) 30.56

Table 2: **Comparison to existing post-processing calibration methods.** NorCal outperforms the calibration methods from [6] and weight normalization from [25].

Segmentation on v1	AP	AP_r	AP_c	AP_f
RFS [11]	22.58	12.30	21.28	28.55
w/ HistBin [57]	21.82	11.28	20.31	28.13
w/ BBQ (AIC) [32]	22.05	11.41	20.72	28.21
w/ Beta calibration [22]	22.55	12.29	21.27	28.49
w/ Isotonic reg. [58]	22.43	12.19	21.12	28.41
w/ Platt scaling [35]	22.55	12.29	21.27	28.49
w/ NorCal	25.22	19.33	24.16	28.98
Detection on v0.5	AP^b	AP^b_r	AP^b_c	AP^b_f
Faster R-CNN	20.98	4.13	19.70	29.30
w/ τ-norm-select [25]	21.61	6.18	20.99	28.54
w/ NorCal	23.87	6.98	24.17	30.24

Table 3: **Ablation studies of NorCal with various modeling choices and mechanisms.** We report results on LVIS v1 instance segmentation. Cal: calibration mechanism. Nor: class score normalization. The best ones are bold.

a_c	Cal	Nor	AP	AP_r	AP_c	AP_f
Baseline	$\exp(\phi_c(\boldsymbol{x}))$	✓	22.58	12.30	21.28	28.55
$\frac{1-\gamma^{N_c}}{1-\gamma}$ (ENS [5])	$\exp(\phi_c(\boldsymbol{x}))/a_c$	✗	23.66	14.55	22.36	**29.11**
		✓	23.96	15.84	22.61	29.04
	$p(c\|\boldsymbol{x})/a_c$	✗	24.18	18.88	22.95	27.87
		✓	24.85	19.43	23.67	28.54
	$\exp(\phi_c(\boldsymbol{x}))/a_c$	✗	17.49	14.16	17.20	19.27
		✓	24.85	**19.43**	23.67	28.54
N_c^γ (CDT [55])	$\exp(\phi_c(\boldsymbol{x}))/a_c$	✗	17.52	14.04	17.38	19.20
		✓	24.77	17.99	23.81	28.83
	$p(c\|\boldsymbol{x})/a_c$	✗	24.50	18.34	23.42	28.41
		✓	25.22	19.33	24.16	28.98
	$\exp(\phi_c(\boldsymbol{x}))/a_c$	✗	17.52	13.93	17.24	19.41
		✓	**25.22**	19.33	**24.16**	28.98

25] which consider cross-category re-ranking, NorCal re-orders the scores within each class. We provide more detailed discussions in ablation studies.

Comparison to existing post-calibration methods. We then compare our method against other post-calibration techniques, first to those in [6] on the LVIS v1 instance segmentation task, including Histogram Binning [57], Bayesian binning into quantiles (BBQ) [32], Beta calibration [22], Isotonic regression [58], and Platt scaling [35], and then to τ-normalization [20, 25] on the LVIS v0.5 object detection task.

Table 2 shows the results. NorCal is significantly superior to other techniques on both tasks and can improve both rare objects and frequent ones. As mentioned in Section 3.2, we believe that part of the reason calibration methods in [6] perform poorly is their sensitivity to hyper-parameter tuning; we instead calibrate each class using the empirical class frequencies, followed by score normalization.

4.3 Ablation studies and analysis

Effect of calibration mechanism. In addition to logit reducing, *i.e.*, scaling on exponentials ($\exp(\phi_c(\boldsymbol{x}))/a_c$ in Eq. 4), we investigate another two types for score calibration methods. Specif-

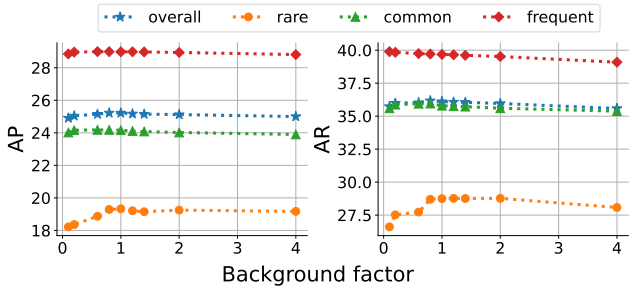

Figure 4: **Results of precision and recall by adjusting background class scores.** Results are on v1 instance segmentation.

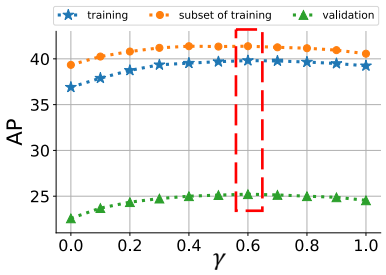

Figure 5: **Calibration factor γ can be robustly tuned with training data.**

ically, we scale down the output logits from the network ($\phi_c(\boldsymbol{x})/a_c$) or the probabilities from the classifier ($p(c|\boldsymbol{x})/a_c$). Again, we keep the background class intact and apply score normalization. In Table 3, we see that calibration on exponentials and probabilities perform the same[4] and outperform on logits, which might be hard to handle the case where c has negative logit $\phi_c(\boldsymbol{x})) < 0$. In contrast, exponentials are guaranteed to be positive and monotonically increasing.

Effect of calibration factors a_c. Beyond the class-dependent temperature (CDT) [55], we provide an alternative of calibration factor a_c, inspired by the effective number of samples (ENS) [5]. Specifically, we experiment $a_c = (1 - \gamma^{N_c})/(1 - \gamma)$. Same as CDT, ENS has a single hyper-parameter γ that controls the degree of dependency between a_c and N_c. If $\gamma = 0$, we recover the original confidence scores. We report the comparison results of those two calibration factors in Table 3. With appropriate post-calibration mechanisms, both can provide consistent improvement over the baseline model.

Importance of class score normalization. Again in Table 3, we compare NorCal with or without class score normalization. By applying normalization, we see that NorCal can improve all categories, including frequent objects. Moreover, it is applicable to different types of post-processing mechanisms as well as calibration factors. We attribute that the normalization operation in Eq. 2 sums the class scores (including the background class) to 1, which is the key to make NorCal excel in AP, especially for frequent objects.

How to handle the background class? NorCal does not calibrate the background classification score. We ablate this design by multiplying the exponential of the background logit with a background calibration factor β, *i.e.*, $\exp(\phi_{C+1}(\boldsymbol{x})) \times \beta$. If $\beta = 1$, there is no any calibration on background class. Figure 4 shows the average precision and recall of the model with NorCal w.r.t different β. When reducing the β, we see the performance drop on both precision and recall, especially on rare classes. This justifies one key ingredient of NorCal: the foreground vs. background classifier (cf. Eq. 3) already performs fairly well when keeping the background class intact.

Sensitivity to the calibration factor. NorCal has one hyper-parameter: the calibration factor γ, that controls the strength of the logit reducing term. We find that this can be tuned easily, even on a 5K subset of *training* images; in Figure 5, we observe that the AP trends on the training and validation sets at different γ's are close to each other. We find this observation is valid among all different models and architectures.

NorCal reduces false positives and re-ranks predictions within each class. In Table 4, we show that NorCal can improve the AR for all classes but slightly degrades frequent objects. Given its improvement on AP, suggesting that NorCal removes false positive and can re-rank the proposals within each class.

NorCal is robust to AP^{Fixed}. Table 5 reports results on AP^{Fixed}. NorCal can further improve the AP^{Fixed} significantly, suggesting that NorCal improves AP due to how it ranks predictions not only across but also within classes.

Table 4: **NorCal can improve precision and recall.** Results are reported on LVIS v1 instance segmentation.

	AP	AP_r	AP_c	Ap_f
RFS [11]	22.58	12.30	21.28	28.55
w/ NorCal	25.22	19.33	24.16	28.98
	AR	AR_r	AR_c	AR_f
RFS [11]	30.61	13.73	28.64	40.24
w/ NorCal	36.10	28.75	35.79	39.68

[4]With class score normalization, they are mathematically the same.

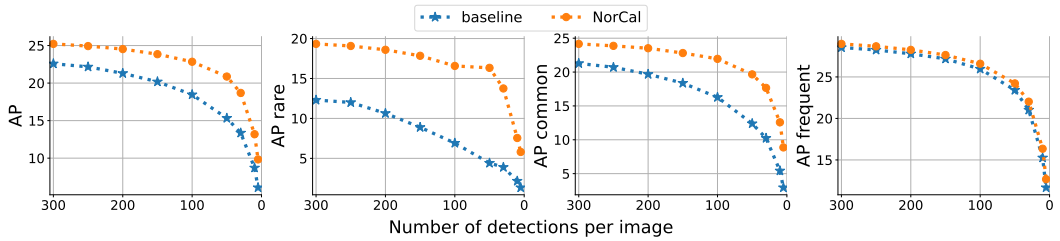

Figure 6: **Limits on #detections per image.** Reducing number of detections per image indicates the quality of confident predictions in the ranking. NORCAL shows better capability to preserve true positives when reducing the cap. It performs better with 100 detections than baseline with 300 without sacrificing frequent classes.

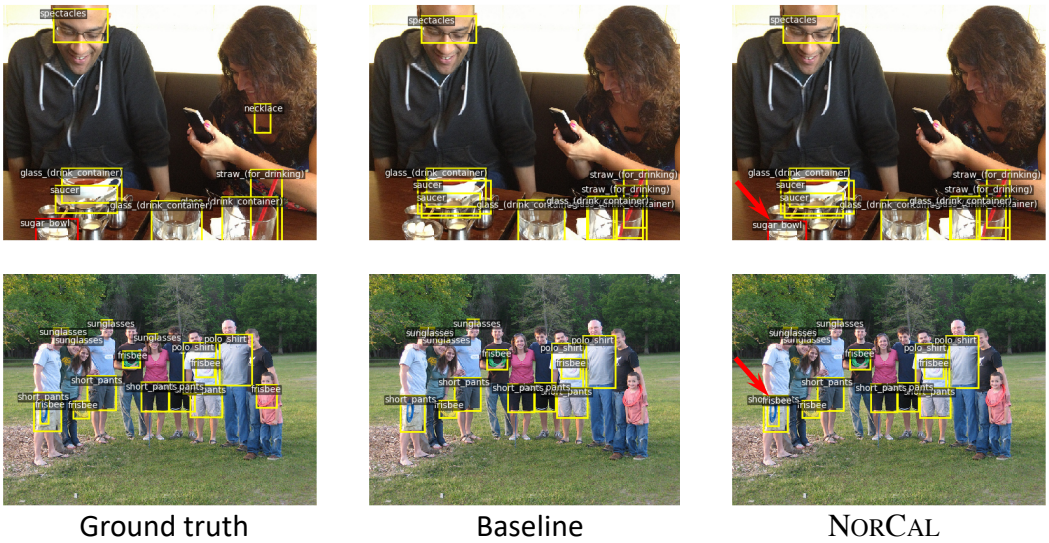

Figure 7: **Qualitative results.** We superimpose red arrows to show the improvement, and Yellow and red boxes to indicate the ground truth labels of frequent and rare classes. In the first example, NORCAL successfully detects the rare object *sugar bowl* without sacrificing other predictions. In the second example, even surprisingly, it can recover the low confident frequent class *frisbee* as well.

Limiting detections per image. Finally, we evaluate the NORCAL by changing the limit on number of detections per image. In Figure 6, NORCAL shows superior performance against the baseline model under all settings. It is worth noting that NORCAL achieves better performance even using a strict 100 #det/img than the baseline model with 300 limit.

Table 5: NORCAL **can improve the baseline model with AP^{Fixed}.** Results are reported on LVIS v1 instance segmentation.

AP Fixed	AP	AP_r	AP_c	AP_f
RFS [11]	25.68	20.07	24.82	29.11
w/ NORCAL	26.26	20.56	25.39	29.73

Qualitative results. We show qualitative bounding box results on LVIS v1 in Figure 7. We compare the ground truth, the results of the baseline and of NORCAL. We observe that NORCAL not only can accurately recognize more objects from rare categories that may be overlooked by the baseline detector, but also improves the detection results on frequent objects. For the upper example in Figure 7, NORCAL discovers a rare object "sugar bowl" without sacrificing any other frequent objects. Moreover, NORCAL can improve the frequent classes, as shown in the bottom example of Figure 7.

5 Conclusion

We present a post-processing calibration method called NORCAL for addressing long-tailed object detection and instance segmentation. We conduct extensive experiments to demonstrate the effectiveness of our method in diverse settings, as well as to validate our modeling choices and analyze our method's mechanisms. By design, our method is simple and does not require re-training of the already-trained models. It is also easily tuned and extensible. We believe that our results indicate that a general useful tool that has been overlooked can give rise to strong baselines for long-tailed object detection and segmentation.

References

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 1

[2] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018. 1, 4

[3] Nadine Chang, Zhiding Yu, Yu-Xiong Wang, Anima Anandkumar, Sanja Fidler, and Jose M Alvarez. Image-level or object-level? a tale of two resampling strategies for long-tailed detection. In *ICML*, 2021. 2

[4] Liang-Chieh Chen, Alexander Hermans, George Papandreou, Florian Schroff, Peng Wang, and Hartwig Adam. Masklab: Instance segmentation by refining object detection with semantic and direction features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4013–4022, 2018. 1

[5] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, 2019. 1, 2, 4, 6, 7, 8

[6] 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, 3, 4, 5, 6, 7

[7] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, 2009. 1

[8] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *CVPR*, 2021. 3

[9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1

[10] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017. 3

[11] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 1, 2, 4, 6, 7, 8, 9

[12] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *European conference on computer vision*, pages 345–360. Springer, 2014. 1

[13] Abdul Mueed Hafiz and Ghulam Mohiuddin Bhat. A survey on instance segmentation: state of the art. *International Journal of Multimedia Information Retrieval*, pages 1–19, 2020. 1

[14] Junwei Han, Dingwen Zhang, Gong Cheng, Nian Liu, and Dong Xu. Advanced deep-learning techniques for salient and category-specific object detection: a survey. *IEEE Signal Processing Magazine*, 35(1): 84–100, 2018. 1

[15] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009. 1, 4

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 1, 2, 6

[17] Ting-I Hsieh, Esther Robb, Hwann-Tzong Chen, and Jia-Bin Huang. Droploss for long-tail instance segmentation. In *AAAI*, 2021. 2, 5

[18] Xinting Hu, Yi Jiang, Kaihua Tang, Jingyuan Chen, Chunyan Miao, and Hanwang Zhang. Learning to segment the tail. In *CVPR*, 2020. 1, 2

[19] Licheng Jiao, Fan Zhang, Fang Liu, Shuyuan Yang, Lingling Li, Zhixi Feng, and Rong Qu. A survey of deep learning-based object detection. *IEEE Access*, 7:128837–128868, 2019. 1

[20] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *ICLR*, 2020. 1, 2, 3, 4, 5, 6, 7

[21] Byungju Kim and Junmo Kim. Adjusting decision boundary for class imbalanced learning. *IEEE Access*, 8:81674–81685, 2020. 1, 3, 4, 5

[22] Meelis Kull, Telmo Silva Filho, and Peter Flach. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In *Artificial Intelligence and Statistics*, pages 623–631. PMLR, 2017. 3, 7

[23] Meelis Kull, Miquel Perello-Nieto, Markus Kängsepp, Hao Song, Peter Flach, et al. Beyond temperature scaling: Obtaining well-calibrated multiclass probabilities with dirichlet calibration. In *NIPS*, 2019. 3

[24] Yin Li, Xiaodi Hou, Christof Koch, James M Rehg, and Alan L Yuille. The secrets of salient object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 280–287, 2014. 1

[25] Yu Li, Tao Wang, Bingyi Kang, Sheng Tang, Chunfeng Wang, Jintao Li, and Jiashi Feng. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In *CVPR*, 2020. 1, 2, 3, 5, 7

[26] 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, 4, 6

[27] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 1, 6

[28] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 1

[29] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *International journal of computer vision*, 128(2):261–318, 2020. 1

[30] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 1

[31] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *ICLR*, 2021. 1, 2, 3, 4

[32] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015. 3, 7

[33] Constantine P Papageorgiou, Michael Oren, and Tomaso Poggio. A general framework for object detection. In *ICCV*, 1998. 1

[34] Sida Peng, Wen Jiang, Huaijin Pi, Xiuli Li, Hujun Bao, and Xiaowei Zhou. Deep snake for real-time instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8533–8542, 2020. 1

[35] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999. 3, 7

[36] Vignesh Ramanathan, Rui Wang, and Dhruv Mahajan. DLWL: Improving detection for lowshot classes with weakly labelled data. In *CVPR*, 2020. 3

[37] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, 2017. 1

[38] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 1

[39] Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. In *NeurIPS*, 2020. 1, 2, 5

[40] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6): 1137–1149, 2016. 1, 2

[41] Li Shen, Zhouchen Lin, and Qingming Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *ECCV*, 2016. 2

[42] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *CVPR*, 2020. 1, 2, 5, 6, 7

[43] Jingru Tan, Gang Zhang, Hanming Deng, Changbao Wang, Lewei Lu, Quanquan Li, and Jifeng Dai. 1st place solution of lvis challenge 2020: A good box is not a guarantee of a good mask. *arXiv preprint arXiv:2009.01559*, 2020. 1

[44] Jingru Tan, Xin Lu, Gang Zhang, Changqing Yin, and Quanquan Li. Equalization loss v2: A new gradient balance approach for long-tailed object detection. In *CVPR*, 2021. 1, 2, 5

[45] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *NeurIPS*, 2020. 3

[46] Junjiao Tian, Yen-Cheng Liu, Nathan Glaser, Yen-Chang Hsu, and Zsolt Kira. Posterior re-calibration for imbalanced datasets. In *NeurIPS*, 2020. 3

[47] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *CVPR*, pages 8769–8778, 2018. 1

[48] Paul Viola, Michael Jones, et al. Robust real-time object detection. *International journal of computer vision*, 4(34-47):4, 2001. 1

[49] Jiaqi Wang, Wenwei Zhang, Yuhang Zang, Yuhang Cao, Jiangmiao Pang, Tao Gong, Kai Chen, Ziwei Liu, Chen Change Loy, and Dahua Lin. Seesaw loss for long-tailed instance segmentation. In *CVPR*, 2021. 1, 2

[50] Tao Wang, Yu Li, Bingyi Kang, Junnan Li, Junhao Liew, Sheng Tang, Steven Hoi, and Jiashi Feng. The devil is in classification: A simple framework for long-tail instance segmentation. In *ECCV*, 2020. 1, 2, 4

[51] Tong Wang, Yousong Zhu, Chaoyang Zhao, Wei Zeng, Jinqiao Wang, and Ming Tang. Adaptive class suppression loss for long-tail object detection. In *CVPR*, 2021. 5

[52] Xin Wang, Thomas E Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. In *ICML*, 2020. 1, 3

[53] 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*, 2020. 1

[54] Tong Wu, Ziwei Liu, Qingqiu Huang, Yu Wang, and Dahua Lin. Adversarial robustness under long-tailed distribution. *arXiv preprint arXiv:2104.02703*, 2021. 1, 4

[55] Han-Jia Ye, Hong-You Chen, De-Chuan Zhan, and Wei-Lun Chao. Identifying and compensating for feature deviation in imbalanced deep learning. *arXiv preprint arXiv:2001.01385*, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[56] Han-Jia Ye, De-Chuan Zhan, and Wei-Lun Chao. Procrustean training for imbalanced deep learning. *arXiv preprint arXiv:2104.01769*, 2021. 5

[57] Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Icml*, volume 1, pages 609–616. Citeseer, 2001. 3, 7

[58] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699, 2002. 3, 7

[59] Yuhang Zang, Chen Huang, and Chen Change Loy. FASA: Feature augmentation and sampling adaptation for long-tailed instance segmentation. *arXiv preprint arXiv:2102.12867*, 2021. 3

[60] Cheng Zhang, Tai-Yu Pan, Yandong Li, Hexiang Hu, Dong Xuan, Soravit Changpinyo, Boqing Gong, and Wei-Lun Chao. MosaicOS: a simple and effective use of object-centric images for long-tailed object detection. *arXiv preprint arXiv:2102.08884*, 2021. 3, 6, 7

[61] Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *CVPR*, 2021. 3

[62] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Joint coco and lvis workshop at eccv 2020: Lvis challenge track technical report: Centernet2. 1

[63] Xiangxin Zhu, Dragomir Anguelov, and Deva Ramanan. Capturing long-tail distributions of object subcategories. In *CVPR*, pages 915–922, 2014. 1

[64] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*, 2019. 1

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Please be referred to the broader impact section in the supplementary material
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See supplementary materials
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No] We use the datasets released for scientific research
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]