Foreground and Background Separate Adaptive Equilibrium Gradients Loss for Long-Tail Object Detection*

Tianran Hao[1[0000−0002−3416−464X]], Ying Tao[1[0000−0001−5804−1081]], Meng Li[1[0000−0002−2939−2664]], Xiao Ma[1[0000−0003−0678−2296]], Peng Dong[1[0000−0002−9338−7704]], Lisha Cui[1[0009−0002−8570−5482]], Pei Lv[1[0000−0002−2654−0561]], and Mingliang Xu[1[0000−0002−6885−3451]]

Zhengzhou University, Zhengzhou 450001, Henan, China

Abstract. The current mainstream object detection methods usually tend to implement on datasets where the categories remain balanced, and have made great progress. However, in the presence of long-tail distribution, the performance is still unsatisfactory. Long-tail data distribution means that a few head classes occupy most of the data, while most of the tail classes are not representative, and tail classes are excessive negatively suppressed during training. Existing methods mainly consider suppression from negative samples of the tail classes to improve the detection performance of the tail classes, while ignoring suppression from correct background prediction. In this paper, we propose a new Foreground and Background Separate Adaptive Equilibrium Gradients Loss for Long-Tail Object Detection (FBS-AEGL) to deal with the problem mentioned above. Firstly, we introduce the numerical factor among categories to weight different classes, then adaptively leverage the suppression of head classes according to the logit value of the network output. Meanwhile, dynamically adjusting the suppression gradient of the background classes to protect the head and common classes while improving the detection performance of the tail classes. We conduct comprehensive experiments on the challenging LVIS benchmark. FBS-AEGL Loss achieved the competitive results, with 29.8% segmentation AP and 29.4% box AP on LVIS v0.5 and 28.8% segmentation AP and 29.4% box AP on LVIS v1.0 based on ResNet-101.

Keywords: Long-tail distribution · Object detection · Re-weighting · Equilibrium gradients.

1 Introduction

Object detection is one of the most representative and challenging tasks in computer vision and plays a central role in other related tasks. Most datasets for

* This work was supported in part by the Zhengzhou Major Science and Technology Project under Grant 2021KJZX0060-6, in part by China Postdoctoral Science Foundation under Grant 2021TQ0301, and in part by the National Natural Science Foundation of China under Grant 62372415, 62036010, 62106232.

general-purpose object detection, such as PASCAL VOC [6] and MS COCO [17], are large-scale and manually balanced by collecting common classes, each with a large number of annotations. In realistic scenarios and practical applications, the data usually shows a long-tail distribution [23], and the detection performance of the tail class decreases rapidly.

The reason for this phenomenon is that the performance of deep learning based methods is built on a large amount of data. The data amount of head classes occupies a large proportion of the whole dataset, while that of tail classes can not enable the model to be trained adequately. Moreover, during the training process, other classes often become the negative samples of the tail class, so the gradient of positive and negative samples received by the tail classes are usually in an imbalanced condition and judged as the incorrect classes. In this situation, the performance of using the traditional object detector will be greatly affected, and the prediction results will be more biased towards the head classes.

To overcome the impact generated by this imbalance data distribution, some researchers have proposed re-sampling and re-weighting strategies to address the long-tail distribution. Existing methods use re-sampling [7, 10, 24, 32, 33] to re-balance the dataset and manually change the distribution of the long tail classes of the datasets. Re-weighting methods [31, 28, 27, 12] rebalance the classes by tuning the weights of different classes during the training process. While all these methods improve results to varying degrees, re-sampling may risk over-fitting and under-fitting. Re-weighting methods primarily focus on suppressing negative gradients from incorrect foreground classification, overlooking the importance of negative gradients from correct background classification. In the background case, the loss received by the classification branch suppresses all foreground class prediction scores. Depicted in Figure 1, we study the effect of such discouraging gradients on the different categories of a long-tail dataset. The curve indicates the ratio of negative gradients generated by the incorrect foreground classes to ground-truth background. A smaller ration represents that background produces more negative gradients. We find that discouraging gradients from background classification contribute a much highter

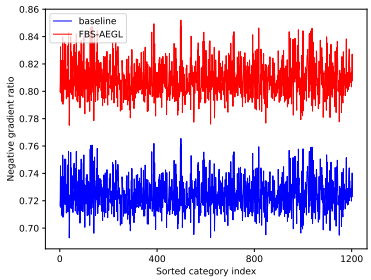

Fig. 1: Percentage of accumulative negative gradients for baseline (blue) and FBS-AEGL (red). The x-axis is the sorted category index of 1203 categories of the LVIS v1.0 dataset. The y-axis is the ratio of accumulative negative gradient for foreground classification and background anchors. We found that the percentage of negative gradients from background is much higher compared to the incorrect foreground prediction.

percentage of total discouraging gradients compared to that of incorrect foreground prediction. While re-weighting methods cannot balance multiple factors simultaneously.

In this paper, we propose a foreground and background separate adaptive equilibrium gradients loss (FBS-AEGL) to address above observed problems. FBS-AEGL mainly relies on the weight factor to regulate the learning process of the object detection network. To clearly demonstrate how the FBS-AEGL works in long-tail object detection, we incorporate FBS-AEGL into a two-stage detector, Mask R-CNN, as our baseline in Figure 2. For the proposals of foreground regions, we first evaluate the quality of samples in each class of the dataset to calculate the number of effective samples in each class. The loss in the prior data volume is rebalanced and reweighted according to the maximum marginal benefit that can be extracted by the network. Then, during the learning process of the model, the logit value of the network output is compared, and a reasonable threshold is set to choose a better sampling level by targeted suppression of the head classes rather than suppressing all negative samples in the tail classes.

Foreground weight factors can effectively mitigate the suppression of classes from foreground region proposals. Considering the balance of gradients for the background region proposals is also essential, we propose background weighting factor for the foreground region proposals. In detail, the Bernoulli distribution is introduced to combine the background weighting factor and foreground weighting factor, and allow the network to fine-tune the overall weights in a stochastic manner. This combination of weighting factors by FBS-AEGL can effectively improve the attention of the network to the tail classes and protect the performance of the head and common classes from excessive suppression while improving the performance of the tail classes.

The proposed FBS-AEGL is trained on two versions of LVIS dataset and evaluated accordingly. Comprehensive experiments demonstrate the effectiveness of FBS-AEGL, with a more competitive performance relative to previous methods. Our main contributions are as follows:

- We propose a new loss function that firstly adopts the weighting factors based on the learning state output of the network and the quality of the dataset, which can efficiently deal with the long-tail object detection problem.
- The loss function is further extended by equalizing the negative gradients generated in the background class, which improves the prediction results of those negative samples in the tail classes.
- We conduct the experiments on the LVIS dataset with a long-tail distribution, and achieve significant improvements, which validate the effectiveness of our method.

2 Related works

2.1 General object detection

With the emergence of convolutional neural networks, many deep learning-based object detection algorithms have achieved quite good detection results. In gen-

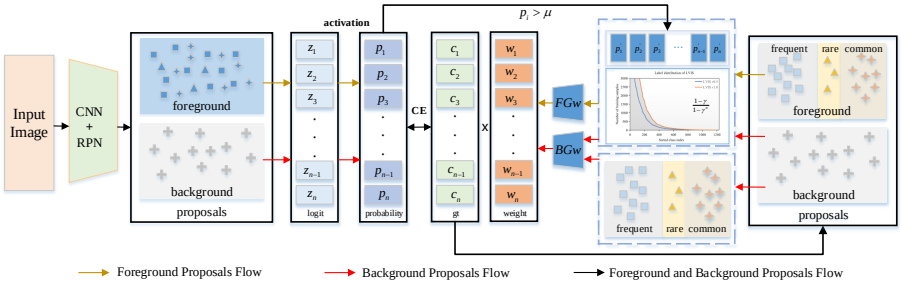

Fig. 2: An illustration of our proposed adaptive foreground and background class suppression loss with two-stage detector. For the proposals, different shapes represent different categories, and different colors of the same category represent different stages. The logit represents the output of the network. Different reweighting strategies for the foreground and background proposals are adopted, which are FGw and BGw, respectively. Activation means the activation function, such as sigmoid and softmax. CE denotes the cross-entropy function.

eral, deep learning-based object detection algorithms can be divided into tow-stage [8, 25, 11, 2] and one-stage methods [16, 18, 22]. The key difference between the two methods is that the two-stage algorithm needs to generate a proposal (a pre-selected box that could potentially contain the object to be detected) and then perform fine-grained object detection, while the one-stage algorithm extracts feature directly in the network to predict object classification and location.

Two-stage methods. The regional-based convolutional neural network (R-CNN) series of work is the most representative work of two-stage object detection methods. The R-CNN [9] first generates 2,000 candidate detections that are most likely to be objects using a selective search method [29], then extracts the depth features of these candidate detection using a deep convolutional neural network, and finally performs classification and regression using a support vector machine. Since R-CNN extracts the depth features of each candidate detection separately, it suffers from the problem of slow inference speed. Fast R-CNN[8] is an improved work of R-CNN. Fast R-CNN first extracts the depth features of the whole image, then scales the features of the candidate detection to a fixed size using region of interest (RoI) pooling operation, and finally performs classification and regression using the fully connected layer. Unlike the previous Fast R-CNN [8], which relies on selective search algorithm, Faster R-CNN [25] introduces the Region Proposal Network (RPN), and unifies the generation of candidate windows with the classification and regression of candidate windows into a single network for learning together. Via adding a mask prediction branch in the Faster R-CNN architecture, Mask R-CNN [11] bridges the gap between object detection and instance segmentation. Different with the two-stage meth-

ods, the one-stage methods does not need to get the proposal box stage and directly generates the class probability and position coordinate values of the object, so it has a faster detection speed.

One-stage methods. The one-stage object detection algorithms are represented by YOLO [20–22, 30] series and SSD [18], which has gone through several iterations. In particular, YOLOv5 provides a variety of object detectors of different sizes to satisfy the needs of different applications, and has been widely used in real-life. SSD uses feature maps of different scales to perform detection, with large scale feature maps for detecting small objects and small scale feature maps for detecting large objects. In general, two-stage methods are more precise in detection, while one-stage methods are faster in inference.

2.2 Long-Tail object detection

Re-sampling. Re-sampling [26, 10, 32, 33] is a most intuitive solution by randomly duplicating more target data from the tail classes for training or removing a certain amount of target data from head classes to tackle the long-tail distribution problem. While those methods achieve significant improvement, they may still have significant over-fitting problems among them. Other works [7, 24, 34] balance data distribute through meta-learning or memory augmentation. For example, they introduce a new quality ranking of candidate regions to enhance the datasets.

Re-weighting. Another typical strategy re-weighting [28, 27, 5] is to give different weights to different classes by the loss function, giving a relatively high weight to the loss of the tail class to expand the impact on the training samples of the tail class, or more fine-grained adjustment at the sample level by multiplying different weights on different training samples to reweight the network loss at the category level. Seesaw loss [31] dynamically rebalances the positive and negative gradients of each sample using a mitigation factor and a compensation factor. EFL [14] focuses on the degree of imbalance between positive and negative samples of each category by introducing a category-dependent moderator. However, the above methods do not take into account the different prediction results of the background and foreground classes, and ignore the treatment of the background classes.

Other-methods. GOL [1] points out that the use of Sigmoid or Softmax functions is responsible for the poor performance of long-tail object detection, and the use of Gumbel activation functions that are more suitable for long-tail data distribution. AHRL [13] visualizes the feature representation of each category in the learned feature space to address the long-tail problem in a coarse-to-fine manner. LDAM [3] is minimized based on the marginal generalization bound. Balanced Group Softmax (BAGS) [15] divides classes with similar number of

samples into groups and applies a softmax function to each group, but with inconsistent training between neighboring classes of similar size. NORCAL [19] investigates a post-processing calibration of confidence scores. ROG [35] design a generalized average precision (GAP) lossto explicitly optimize the global-level score ranking across different objects.

Unlike the above methods, our method focuses on both foreground and background categories. We start from the impact of the data volume of the dataset itself on the model, and adaptively adjust the suppression gradient of each class according to the logit value of the model output, so that the model focuses more on the tail class and improves the discriminative ability between semantically similar categories of the head classes and tail ones.

3 Methodology

As mentioned in Section 1, for long-tail object detection, equilibrium gradients from foreground and background region proposals for categories are two intertwined and equally vital parts, while the quantity and difficulty of data is root cause. In this work, we propose Foreground and Background Separate Adaptive Equilibrium Gradients Loss to balance gradient for each category, which consists of two components: 1) adaptive dynamically adjusts gradient for each class from foreground region proposals, 2) adaptive further balances gradient for each class from background region proposals. The proposed FBS-AEGL is flexible enough to be applied to existing detectors.

3.1 Revisiting sigmoid cross-entropy loss

The sigmoid cross-entropy loss is widely adopted in object detection, so we first revisit it:

$$p_i = \frac{1}{1 + e^{-z_i}}, \tag{1}$$

$$L_{BCE} = -\sum_{i=1}^{C} \log\left(\hat{p}_i\right), \quad \hat{p}_i = \begin{cases} p_i, & \text{if } i = c, \\ 1 - p_i, & \text{otherwise,} \end{cases} \tag{2}$$

where C is the number of categories. z_i denotes the logit of the network output of class i. p_i denotes the probability that the current sample belongs to class i as calculated by Equation 1. The gradient of the loss function with respect to z_i is derived as:

$$\frac{\partial L_{BCE}}{\partial z_i} = \begin{cases} p_i - 1, & \text{if } i = c, \\ p_i, & \text{otherwise.} \end{cases} \tag{3}$$

The sigmoid cross-entropy application with long-tail object detection, giving a sample which foreground prediction of category c, for rare categories $i(i \neq c)$, they will receive negative suppression gradients and result in a low probability of network output. Such negative gradients will occur in large numbers from the frequent classes and impede the positive activation of tail classes. On the other hand, the background samples are negative samples for all categories. Our

core idea is to mitigate the negative gradients of each category, both in terms of foreground and background predictions.

3.2 Foreground and background separate adaptive equilibrium gradients loss

In this section, we introduce FBS-AEGL to balance gradients for each category from foreground and background proposals, which considers class sizes and the network learning status.

Formally, we introduce a weight term w to the original sigmoid cross-entropy loss function, and the Foreground and Background Separate Adaptive Equilibrium Gradients Loss as:

$$L_{FBS-AEGL} = -\sum_{i=1}^{C} w_i \log\left(\hat{p}_i\right), \tag{4}$$

$$\hat{p_i} = \begin{cases} p_i, & \text{if } i = c, \\ 1 - p_i, & \text{otherwise.} \end{cases} \tag{5}$$

For a region proposal r, we set w_i with the following as:

$$w_i = \begin{cases} FGw_i, \text{ if } E(r) = 1, \\ BGw_i, \text{ otherwise,} \end{cases} \tag{6}$$

where FGw_i denotes the weight generated by FGw for class i of foreground proposal, BGw_i denotes the weight generated by BGw for class i of background proposal. $E(r)$ indicates whether the proposal is foreground or background. $E(r)$ equals to 1 means r is a foreground region and vice versa is background. FGw and BGw denote the re-weighting strategies for the foreground and background proposals, respectively. The gradient of FBS-AEGL with respect to z_i can be derived as:

$$\frac{\partial L_{FBS-AEGL}}{\partial z_i} = \begin{cases} w_i p_i - 1, \text{ if } i = c, \\ w_i p_i, \quad \text{otherwise.} \end{cases} \tag{7}$$

We will discuss the FGw and BGw in detail, respectively.

Foreground weight (FGw). FGw focuses on foreground region proposals to adjust gradients for each class. First, we adopt the weighting factor $(1-\gamma)/(1-\gamma^n)$ that uses the effective number of actual training instances per category to re-balance the loss from class sizes following CB loss function [5]. Where n denotes the number of instances and $\gamma \in [0, 1)$ is a hyper-parameter. Then, we allow the model to more accurately adjust the loss based on the learning status. Finally, we multiply FGw_i by the loss term $-log(\hat{p}_i)$ for category i. The formulation of FGw_i is designed as:

$$FGw_i = \begin{cases} (1-\gamma)/(1-\gamma^{n_i}), \text{ if } i = c, \\ (1-\gamma)/(1-\gamma^{n_i}), \text{ if } i \neq c \quad and \quad p_i \geq \mu, \\ 0, \qquad\qquad\qquad\quad \text{if } i \neq c \quad and \quad p_i < \mu, \end{cases} \tag{8}$$

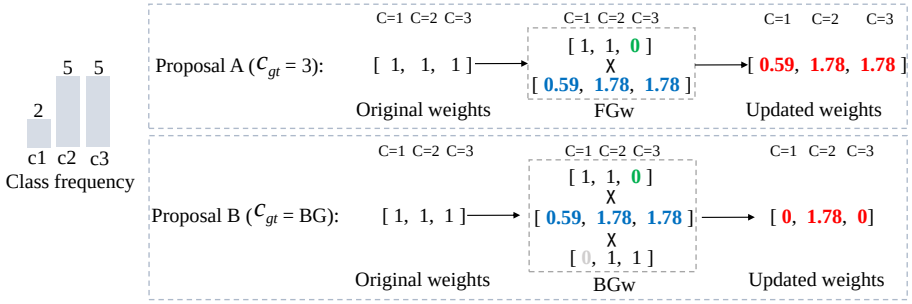

Fig. 3: Illustration of an example which leverages the suppression gradients from our proposed FBS-AEGL. The upper part describes the FGw and the lower part describes the BGw. Here we assume there are three possible foreground classed, and show the ground-truth classes (i.e., c_{gt}) and original weights for foreground proposal and background proposal. Green numbers, blue numbers, and grey numbers are handled by category probabilities, the weighting factor, and Bernoulli, respectively. Red numbers indicate final weights.

where n_i denotes the number of instances belonging to category i in dataset. As category c, FGw_i is set to $(1-\gamma)/(1-\gamma^{n_i})$, if the current proposal belongs to category c. For other categories i ($i \neq c$), we apply the sigmoid probability p_i as a signal to decide whether to suppress category i. The sigmoid probability works well because it does not assume mutual exclusivity between classes and can be a good representation of fine-grained features. If p_i is bigger than μ, that means the network considers category i as similar to c, then we set FGw_i to $(1-\gamma)/(1-\gamma^{n_i})$ for discriminative learning. Otherwise, FGw_i equals to 0, FGw_i will be set as 0 to alleviate needless negative suppression. Our proposed FGw integrates the data quantity and the network output probability to adjust the gradient with each class more precisely. Let us consider a three-class example (see Figure 3), in which c=1 is a tail class, and c=2 and c=3 are head classes. Assuming that foreground proposal A is found from an image: proposal A has original weights [1 ,1 ,1]. Let us suppose c1 = 2, c2 = c3 = 5, and by applying Equation 8, we get the new weights for proposal A as [0.59, 1.78, 1.78], which reduced tail category suppression for c = 1.

Background weight (BGw). FGw effectively alleviates negative suppression gradients for each class from foreground region proposals. However, the gradients balanced from the background region proposals is more important, as they occupy a large proportion of the train learning.

Therefore, we design BGw to mitigate negative suppression gradients from the background proposals. Similar to the design of the FGw, we seek to further optimize to determine the category of suppression. The role of BGw is to mitigate the accumulation of small but non-negligible discouraging gradients from the background. We introduce a Bernoulli distribution to better combine with

FGw, allowing the network to fine-tune the weights in a stochastic manner [12], thus dynamically balancing the effect of background suppression gradients for rare/common/frequent categories. We formulate BGw_i as follows:

$$BGw_i = \begin{cases} w^i_{Ber} \cdot \frac{(1-\gamma)}{(1-\gamma^{n_i})}, & \text{if } p_i \geq \mu, \\ 0, & \text{if } p_i < \mu, \end{cases} \tag{9}$$

$$w^i_{Ber} = \begin{cases} 1, \text{ if } z_i = 1 \quad and \quad Ber(\sigma_{z_i}) < \sigma_{z_i}, \\ 1, \text{ if } z_i \neq 1, \\ 0, \text{ otherwise}, \end{cases} \tag{10}$$

where $w^i_{Ber} \in \{0, 1\}$ which as a signal to decide whether to suppress category i. If the region proposal r belongs to the background, w^i_{Ber} is drawn from Bernoulli distribution $Ber(\sigma_{z_i})$ which denotes the random value of the class i. The parameter σ_{z_i} determined by the number of low-frequency categories in the current batch of region proposals. If z_i equals to 1 means the network output logit of category i belong to low frequency category, $\sigma_{z_i} = (n_r + n_c)/n_{all}$, else, z_i not equals to 1 means the network output logit of category i belongs to frequent category, $\sigma_{z_i} = (n_f)/n_{all}$, where n_r, n_c, and n_f indicate the number of rare, common, and frequent foreground region proposals in the current training batch, respectively. n_{all} indicates the total number of foreground region proposals, which is equal to the sum of n_r, n_c, and n_f. As shown in the lower part of Figure 3, assuming that background proposal B is found from an image: proposal B has initial weights [1, 1, 1], by applying Equation 9, we get the new weights for proposal B as [0, 1.78, 0] that eliminate the suppression of the tail category.

In summary, we design FGw and BGw strategies for the foreground and background, respectively, which takes into class sizes and the network learning status to balance gradients for each category.

4 Experiments on LVIS

4.1 Datasets and evaluation metric

We perform experiments on the long-tail and large-scale dataset LVIS [10], which has accurate bounding box and mask annotations each categories. We mainly conduct experiments on the challenging LVIS v1.0 dataset that contains 1203 categories. We train our model on the training set (100K images) and evaluate it on the validation set (19.8K images). LVIS counts the number of images in each category and then divides all categories into three groups: frequent category with more than 100 images, common category with 11-100 images, and rare category with less than 10 images. In addition to the widely used IoU threshold (0.5 - 0.95) for the metric AP, LVIS also reports AP_r (rare category), AP_c (common category), and AP_f (frequent category) to portray the performance of long-tail classes. Like most existing works, we have experimented predominantly with the LVIS v1.0 dataset and present extra key results on LVIS v0.5 dataset. As shown in Figure 4, we visualize the number of training instances for categories in LVIS v0.5 and v1.0 training set.

4.2 Implementation details

For our experiments, we choose Mask R-CNN detector with FPN structure. To compare with the state-of-the-art methods, we also employ Mask R-CNN on LVIS v0.5 and v1.0 datasets, using different backbones, in combination with our proposed FBS-AEGL. The ResNet backbone is initialized by the ImageNet pre-training model. During the training phase, scale jitter and random horizontal flipping are adopted as the default data augmentation. We use 4 GPUs (NVIDIA Tesla V100) with a batch size of 16 (4 images on each GPU). The optimizer is set to stochastic gradient descent (SGD) with 0.9 momentum and 0.0001 weight decay. The initial learning rate is set to 0.02 and is warmed up with 500 iterations. The learning rate decays to 0.002 at epoch 16 and to 0.0002 at epoch 22. The total number of training epochs is 24. During the inference phase, we first resize the images used to shortside of 800 pixels and longside of no more than 1333 pixels. We begin by applying Non-Maximal Suppression, with an IoU threshold of 0.5, to eliminate duplicate items. After that the first 300 detections will be chosen to be final result. The other hyper-parameter settings, such as anchor scale and anchor ratio, are consistent with the same default settings in MMDetection [4]. We concentrate on the classification subnetwork of the Mask R-CNN in our experiments using FBS-AEGL Loss and replace the original softmax cross-entropy loss using our proposed loss function for long-tail datasets with repeat factor sampling (RFS) [10].

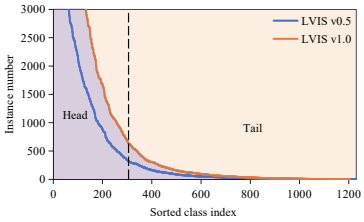

Fig. 4: Statistics of instance number for each category on LVIS v0.5 and v1.0 training set.

4.3 Ablation studies

In order to better analyze FBS-AEGL, we set up the following groups of ablation experiments:

Effectiveness of FGw and BGw. In addition to this, we perform ablation experiments to verify the performance of the core strategy in our method with Mask R-CNN ResNet-50-FPN backbone. FGw, BGw denote foreground and background suppression factor, respectively. FGw focuses on foreground region proposals to adjust gradients for each class. BGw effectively alleviates negative suppression gradients each class from background region proposals. As shown in Table 1, the 6.4% box AP and 9.3% box AR improvement is achieved on the network with FGw, BGw which demonstrate the effectiveness of its two factors.
Sampling factor γ and Suppressing factor μ. FBS-AEGL introduces two hyper-parameters, which are the sampling factor and the suppressing factor. The sampling factor γ defines the effective number of actual training instances.

Table 1: Ablation study of FGw, BGw in FBS-AEGL with Mask R-CNN and ResNet-50-FPN as the backbone for LVIS v1.0 val set. FGw, BGw indicate foreground and background suppression factor respectively.

FGw	BGw	AP^b	AP	AP_r	AP_c	AP_f	AR^b
		21.4	20.5	1.1	18.6	31.0	28.4
√		26.6	26.9	16.2	27.3	29.7	37.0
	√	26.8	27.0	18.0	26.6	31.0	37.1
√	√	**27.8**	**27.6**	**19.1**	**27.0**	**32.5**	**37.7**

The suppressing factor μ indicates the degree of suppression for each category which is a trade-off between reliving over-suppression on tail classes and chasing discriminative learning. A small μ means that most of the categories will be suppressed, which will suppress too much on tail categories. However, for an extremely large μ, the network will only suppress categories with extremely high confidences while ignore most of the other categories, thus will weaken the classifier's discriminative power. For all experiments we mainly use ResNet-50 Mask R-CNN and LVIS v1.0 dataset. To explore how γ and μ influence the predicted results, we experiment with several different values and the results are reported in Table 2. As shown in Table 2, both the sampling factor and the suppressing factor have played an essential role in FBS-AEGL. Through the two components working in synergy, FBS-AEGL dramatically improves the performance of improved baseline from 21.4% box AP to 27.8% box AP. We empirically find $\gamma = 0.7$ and $\mu = 0.7$ works best under current setting.

Table 2: Ablation study of the hyper-parameter γ and μ.

μ	γ	AP^b	AP	AP_r	AP_c	AP_f
0.01	0.7	25.1	25.6	16.0	24.3	31.3
0.1	0.7	25.6	25.2	17.7	24.4	30.4
0.3	0.7	26.7	26.7	18.9	25.8	31.2
0.5	0.7	27.3	27.1	19.4	26.6	31.0
0.7	0.7	**27.8**	**27.6**	19.1	27.0	**32.5**
0.9	0.7	27.1	27.0	17.7	26.3	32.2
0.7	0.5	27.4	27.1	18.1	27.0	31.2
0.7	0.6	27.6	27.4	19.6	**27.1**	31.2
0.7	0.8	27.7	27.5	**20.4**	27.0	31.2
0.7	0.9	27.6	27.1	18.8	26.8	31.1

Table 3: Comparative results for LVIS v1.0 val set using random sampler and RFS sampler in FBS-AEGL with Mask R-CNN and ResNet-50-FPN as the backbone.

Sampler	AP^b	AP	AP_r	AP_c	AP_f
Random	27.4	27.1	18.1	27.0	31.2
RFS	27.8	27.6	19.1	27.0	32.5

Table 4: Ablation study of the weighting factor in FBS-AEGL with Mask R-CNN and ResNet-50-FPN as the backbone.

Method	AP^b	AP	AP_r	AP_c	AP_f
w/o WF	27.0	27.0	18.1	26.0	31.9
w/ WF	27.8	27.6	19.1	27.0	32.5

Table 5: Comparisons our proposed method plugged into various loss functions for LVIS v1.0 val set. # indicated used RFS sampler.

Method	backbone	AP^b	AP	AP_r	AP_c	AP_f
Mask R-CNN# w/CE	ResNet-50-FPN	24.7	23.7	13.3	23.0	29.0
FBS-AEGL# w/CE	ResNet-50-FPN	27.8	27.6	19.1	27.0	32.5
FBS-AEGL# w/GOL	ResNet-50-FPN	28.0	27.8	21.9	27.9	32.5
Mask R-CNN# w/CE	ResNet-101-FPN	27.0	25.7	17.5	24.6	30.6
FBS-AEGL# w/CE	ResNet-101-FPN	29.4	28.8	21.6	28.4	33.8
FBS-AEGL# w/GOL	ResNet-101-FPN	30.4	30.0	23.2	30.4	34.1

Random Sampler and RFS Sampler. We conduct ablation experiments for different sampling strategies, as shown in Table 3, using random sampler and RFS sampler respectively, and achieved 0.3% box AP improvement using RFS sampler with the same backbone network.

Effectiveness of weighting factor. We perform ablation experiments for the effectiveness of the weighting factor (WF) formula (1-γ)/(1-γ^n), as shown in Table 4 , we first ignore the WF in calculating the FBS-AEGL loss, the statistics of a single batch are not representative of the entire training set. In the face of a long-tailed distribution, it cannot optimize the AP for all categories, which leads to suboptimal results. When WF is introduced based on the predefined category distribution of the dataset, the AP gains brings improvements (0.8 points on AP^b and 1.0 points on AP_r). The experimental results verify the effectiveness of the weighting factor, which achieve AP^b of 27.8% and AP_r of 19.1%.

Effectiveness of FBS-AEGL. Our proposed FBS-AEGL is can be used alongside with other loss functions, we perform experiments plugged into GOL methods to confirm the effectiveness of our approach. As shown in Table 5 , the experimental results validate that our method can achieve better results inserted into GOL method. For the baseline model trained with sigmoid CE loss and RFS sampler using ResNet-50 as the backbone, FBS-AEGL loss could improve the AP of object detection. It is noted that the AP for rare categories rises 5.8%. Furthermore, we apply our FBS-AEGL loss with the GOL. We find that FBS-AEGL loss still brings solid improvements (e.g., +8.6 AP_r). We replace ResNet-50 with ResNet-101. Insertion of FBS-AEGL into GOL method which achieves 30.4% box AP, outperforming other competitive methods including GOL (29.2%), ROG (29.3) by 1.2%, 1.1%, respectively. The experimental results have verified the effectiveness of our method.

4.4 Generalization on stronger models

We perform further experiments by replacing larger backbones in order to confirm the generalization of our approach. We replace ResNet-50 with ResNet-101 and Swin-Transformer. The experimental results are as the concluded in Table 6. The experimental results validate that our method can achieve good results in

better backbone as well. In the case of using ResNet-101 and Swin-Transformer as the backbone, the box AP improves by 8.0% and 10.1%, respectively, compared to the baseline model. In addition, by observing the experimental results, it can also be seen that FBS-AEGL has a huge improvement in handling rare categories under different backbones (e.g., the improved AP_r for ResNet-50 is 19.1%, for ResNet-101 is 21.6% and for Swin-Transformer is 23.0%.), which indicates that FBS-AEGL has an excellent performance in handling long-tail data.

Table 6: Comparisons between our proposed method and the baseline Mask R-CNN based on various backbones for LVIS v1.0 val set.

Method	backbone	AP^b	AP	AP_r	AP_c	AP_f
Mask R-CNN	ResNet-50-FPN	21.4	20.5	1.1	18.6	31.0
FBS-AEGL(ours)	ResNet-50-FPN	**27.8**	**27.6**	**19.1**	**27.0**	**32.5**
Mask R-CNN	ResNet-101-FPN	22.8	21.8	1.4	20.3	32.5
FBS-AEGL(ours)	ResNet-101-FPN	**29.4**	**28.8**	**21.6**	**28.4**	**33.8**
Mask R-CNN	Swin-Transformer	24.6	24.2	2.6	22.5	35.4
FBS-AEGL(ours)	Swin-Transformer	**31.5**	**31.0**	**23.0**	**30.5**	**36.2**

(a) frequent classes (b) common classes (c) rare classes

Fig. 5: The box AP for the baseline and FBS-AEGL on frequent, common, and rare classes, respectively. For both models, the ResNet-50-FPN backbone is used for training. The x-axis represents the sorted class index. The y-axis represents the accuracy.

4.5 Performance analysis

As shown in Figure 5, we exhibit the result of baseline and FBS-AEGL on rare, common, and frequent categories on the LVIS v1.0 dataset. Figure 5(a) displays the AP for the frequent category. The two curves are nearly overlapping each other, indicating that our approach does not compromise the performance of the head category. For the common categories (Figure 5(b)), our method starts to demonstrate advantages and can even detect many categories that the baseline model cannot detect. As shown in Figure 5(c), our performance is significantly better than the baseline. The orange curve (ours) has a considerably larger area of integration than the blue curve (baseline). This indicates that our method protects and even improves the head and common classes, while enhancing the tail classes' performance.

Table 7: Comparison with state-of-the-art methods on LVIS v0.5 and v1.0. All models use Mask R-CNN. * denotes that the experimental results in the table are directly from the reference. # indicates that the experimental results are trained with the RFS sampler.

Method	Backbone	Dataset	AP^b	AP	AP_r	AP_c	AP_f
RFS*# [10]	R-50-FPN	LVIS v0.5	25.4	25.4	16.3	25.7	28.7
EQL* [28]	R-50-FPN	LVIS v0.5	23.3	22.8	11.3	24.7	25.1
SimCal* [32]	R-50-FPN	LVIS v0.5	22.6	23.4	16.4	22.5	27.2
Forest R-CNN* [33]	R-50-FPN	LVIS v0.5	25.9	25.6	18.3	26.4	27.6
BAGS* [15]	R-50-FPN	LVIS v0.5	25.8	26.3	18.0	26.9	28.7
LOCE* [7]	R-50-FPN	LVIS v0.5	28.2	28.4	**22.0**	29.0	30.2
DropLoss* [12]	R-50-FPN	LVIS v0.5	25.1	25.5	13.2	27.9	27.3
EQL v2* [27]	R-50-FPN	LVIS v0.5	27.0	27.1	18.6	27.6	29.9
AHRL*# [13]	R-50-FPN	LVIS v0.5	27.4	27.3	17.5	29.0	29.1
FBS-AEGL(ours)#	R-50-FPN	LVIS v0.5	**28.5**	**28.9**	20.9	**30.0**	**30.6**
DropLoss* [12]	R-101-FPN	LVIS v0.5	26.8	26.9	14.8	29.7	28.3
EQL v2* [27]	R-101-FPN	LVIS v0.5	28.1	28.1	20.7	28.3	30.9
AHRL* [13]	R-101-FPN	LVIS v0.5	29.3	29.1	**21.3**	30.7	30.3
FBS-AEGL(ours)#	R-101-FPN	LVIS v0.5	**29.4**	**29.8**	20.7	**31.0**	**31.9**
RFS*# [10]	R-50-FPN	LVIS v1.0	24.7	23.7	13.5	22.8	29.3
LOCE* [7]	R-50-FPN	LVIS v1.0	27.4	26.6	18.5	26.2	30.7
DropLoss*# [12]	R-50-FPN	LVIS v1.0	22.9	22.3	12.4	22.3	26.5
EQL v2*# [27]	R-50-FPN	LVIS v1.0	26.1	25.5	17.7	24.3	30.2
Seesaw* [31]	R-50-FPN	LVIS v1.0	27.4	26.4	19.6	26.1	29.8
FREESEG* [34]	R-50-FPN	LVIS v1.0	26.0	25.2	20.2	23.8	28.9
AHRL*# [13]	R-50-FPN	LVIS v1.0	26.4	25.7	–	–	–
GOL*# [1]	R-50-FPN	LVIS v1.0	27.5	**27.7**	**21.4**	**27.7**	30.4
ROG*# [35]	R-50-FPN	LVIS v1.0	27.2	26.9	20.1	26.8	30.0
FBS-AEGL(ours)#	R-50-FPN	LVIS v1.0	**27.8**	27.6	19.1	27.0	**32.5**
RFS*# [10]	R-101-FPN	LVIS v1.0	26.6	25.5	16.6	24.5	30.6
LOCE* [7]	R-101-FPN	LVIS v1.0	29.0	28.0	19.5	27.8	32.0
EQL v2* [27]	R-101-FPN	LVIS v1.0	27.9	27.2	20.6	25.9	31.4
Seesaw*# [31]	R-101-FPN	LVIS v1.0	28.9	28.1	20.0	28.0	31.8
FREESEG* [34]	R-101-FPN	LVIS v1.0	28.6	27.5	**23.0**	26.5	30.7
AHRL*# [13]	R-101-FPN	LVIS v1.0	28.7	27.6	–	–	–
GOL*# [1]	R-101-FPN	LVIS v1.0	29.2	**29.0**	22.8	29.0	31.7
ROG*# [35]	R-101-FPN	LVIS v1.0	29.3	28.8	21.1	**29.1**	31.8
FBS-AEGL(ours)#	R-101-FPN	LVIS v1.0	**29.4**	28.8	21.6	28.4	**33.8**

4.6 Comparison with state-of-the-art methods

We compare the proposed FBS-AEGL with Mask R-CNN in our experiments and perform with other competitive methods on LVIS v0.5 and LVIS v1.0, and presents the results in Table 7. For LVIS v0.5, we present the results of Mask R-CNN with ResNet50-FPN backbone. Our method achieves 28.5% box AP and segmentation performance of 28.9%AP, outperforming other competitive methods including EQL v2 [27] (27.0%), LOCE* [7] (28.2%) and AHRL [13]

(27.4%) by 1.5%, 0.3%, and 1.1%, respectively. These results demonstrate that our method effectively protects the performance of the common class (30.0%) and head class (30.6%) while also improving the performance of the tail classes. Notably, other methods were unable to achieve this level of performance protection for both the head and common classes, further validating the effectiveness of our approach. For LVIS v1.0, we present the results of using the ResNet50-FPN and ResNet101-FPN backbones with Mask R-CNN. For ResNet50-FPN, FBS-AEGL achieved the best performance with 27.8% box AP, outperforming other methods such as ROG [35], AHRL [13], and GOL [1]. With the larger ResNet101-FPN backbone, our method achieved the best results with 29.4% box AP and 28.8% segmentation AP. Although FBS-AEGL Loss doesn't achieve the best result of AP_r, it obtains the competitive result on AP_c or AP_f, leading to the highest overall performance. We speculate the reason is that other methods focus on optimizing the performance of the tail categories at the expense of the performance of common and head categories. While our method focuses all categories, thus can achieve the best overall performance.

Table 8: Results on COCO-LT minival set. AP^m and AP^b indicate the Mask mAP and Bbox mAP, respectively. AP^b_1, AP^b_2, AP^b_3, AP^b_4 refer to bin of [1,20), [20,400), [400,8000), [8000,-) training instances.

Method	AP^m_1	AP^m_2	AP^m_3	AP^m_4	AP^m	AP^b_1	AP^b_2	AP^b_3	AP^b_4	AP^b
Mask R-CNN	0.0	8.2	24.4	26.0	18.1	0.0	9.5	27.5	30.3	21.4
SimCal	15.0	16.2	24.3	26.0	21.8	14.5	18.0	27.3	30.3	24.6
FASA	13.5	19.0	25.2	27.5	23.4	-	-	-	-	26.0
FREESEG	15.8	20.6	27.6	28.8	25.1	-	-	-	-	-
FBS-AEGL(ours)	**18.6**	**21.9**	**28.0**	**29.0**	**26.0**	**16.6**	**21.3**	**30.5**	**32.4**	**27.5**

4.7 Evaluation on COCO-LT

To confirm the generalization ability to other datasets, we evaluated FBS-AEGL on COCO-LT dataset [32]. The COCO-LT contains 80 classes and about 100K images. COCO-LT dataset defines four class groups [1, 20), [20, 400), [400, 8000), [8000, -) and reports performance as AP_1, AP_2, AP_3, AP_4. For a fair comparison, we used the same experimental setup as SimCal [32]. As shown in Table 8, FBS-AEGL (with Mask R-CNN as baseline) achieves AP^b of 27.5 with the ResNet-50 backbone, which outperforms SimCal and FASA by 2.9% AP^b and 1.5% AP^b, respectively. And, the rare categories (AP^b_1 and AP^b_2) has also been significantly gains. All experimental results demonstrate the advantages and generalizability of our method.

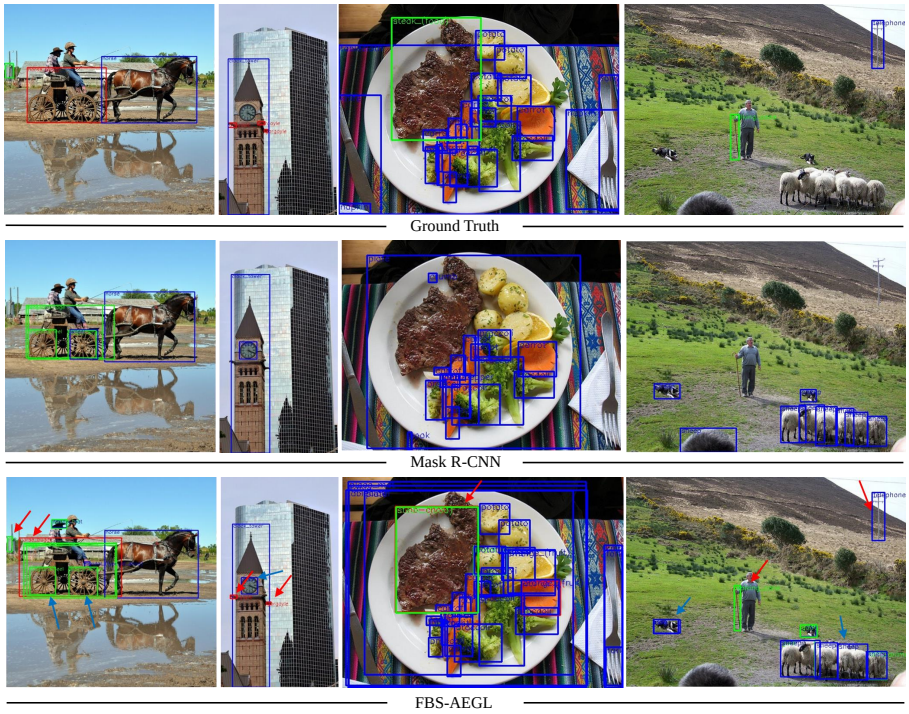

Fig. 6: Prediction results of Mask R-CNN framework without and with FBS-AEGL on the LVIS v1.0 validation set. Compared to the baseline method, our method shows significant improvement on tail, common and frequent classes, and detects many classes that are not detected by the baseline method.(e.g., horse_buggy, gargoyle, silo, steak_(food), walking_cane, telephone). We use red arrows to indicate where we did correct while Mask R-CNN did wrong, and blue arrows to show where we detect while the ground truth is not labeled. (e.g., sunhat, wagon_wheel, dog, sheep). Blue/Green/Red boxes indicate frequent/common/rare category labels.

4.8 Result Visualization

FBS-AEGL not only improves the performance of the tail classes, but also does not compromise the performance of the head classes and additionally detects some classes that were not detected in the baseline. To better interpret the result, we provide qualitative results on LVIS v1.0 in Figure 6. We show the (predicted) bounding boxes from the ground truth annotations, the baseline Mask R-CNN, and FBS-AEGL. We observe that our method can accurately identify more objects in rare and common categories that may be ignored by the baseline detector. For example, FBS-AEGL can correctly detect horse_buggy, gargoyle, silo and steak_(food). They are rare or common categories and the baseline detector fails to make any correct detection on them.

5 Conclusion

In this paper, we propose an foreground and background separate adaptive equilibrium gradients Loss (FBS-AEGL) that is introduced with weight factors based on the learning state output of the network and the quality of the dataset itself. FBS-AEGL mitigates the issue of oversuppression of tail classes by head and common classes in long-tail object detection. This improvement aims to enhance the performance of tail classes without compromising the performance of head and common classes. The proposed method further equalizes against the negative suppression gradients generated by the background class. The experimental results on the long-tail dataset LVIS validate the effectiveness of our method and provide a simple and effective solution for long-tail object detection. A future study will explore the use of the proposed FBS-AEGL for other long-tail distributed vision tasks, such as one/few shot learning, and active learning.

Limitations. In FBS-AEGL, we introduced two hyper-parameters γ and μ that need to be tuned for different datasets. In the future, we plan to extend our method by incorporating reasonable assumptions on the data distribution or designing learning-based, adaptive methods. Currently, we focus on the two-stage detector to address long-tail detection, after which we plan to explore simple and fast one-stage detectors that are widely used in the industry.

References

1. Alexandridis, K.P., Deng, J., Nguyen, A., Luo, S.: Long-tailed instance segmentation using gumbel optimized loss. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X. pp. 353–369. Springer (2022)
2. Cai, Z., Vasconcelos, N.: Cascade r-cnn: High quality object detection and instance segmentation. IEEE transactions on pattern analysis and machine intelligence **43**(5), 1483–1498 (2019)
3. Cao, K., Wei, C., Gaidon, A., Arechiga, N., Ma, T.: Learning imbalanced datasets with label-distribution-aware margin loss. Advances in neural information processing systems **32** (2019)
4. Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., et al.: Mmdetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155 (2019)
5. Cui, Y., Jia, M., Lin, T.Y., Song, Y., Belongie, S.: Class-balanced loss based on effective number of samples. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9268–9277 (2019)
6. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. International journal of computer vision **111**, 98–136 (2015)
7. Feng, C., Zhong, Y., Huang, W.: Exploring classification equilibrium in long-tailed object detection. In: Proceedings of the IEEE/CVF International conference on computer vision. pp. 3417–3426 (2021)

8. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (December 2015)
9. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 580–587 (2014)
10. Gupta, A., Dollar, P., Girshick, R.: Lvis: A dataset for large vocabulary instance segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5356–5364 (2019)
11. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
12. Hsieh, T.I., Robb, E., Chen, H.T., Huang, J.B.: Droploss for long-tail instance segmentation. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 1549–1557 (2021)
13. Li, B.: Adaptive hierarchical representation learning for long-tailed object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2313–2322 (2022)
14. Li, B., Yao, Y., Tan, J., Zhang, G., Yu, F., Lu, J., Luo, Y.: Equalized focal loss for dense long-tailed object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6990–6999 (2022)
15. 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)
16. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
17. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014)
18. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. pp. 21–37. Springer (2016)
19. Pan, T.Y., Zhang, C., Li, Y., Hu, H., Xuan, D., Changpinyo, S., Gong, B., Chao, W.L.: On model calibration for long-tailed object detection and instance segmentation. Advances in Neural Information Processing Systems **34**, 2529–2542 (2021)
20. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
21. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7263–7271 (2017)
22. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
23. Reed, W.J.: The pareto, zipf and other power laws. Economics letters **74**(1), 15–19 (2001)
24. Ren, J., Yu, C., Ma, X., Zhao, H., Yi, S., et al.: Balanced meta-softmax for long-tailed visual recognition. Advances in neural information processing systems **33**, 4175–4186 (2020)

25. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)
26. Shen, L., Lin, Z., Huang, Q.: Relay backpropagation for effective learning of deep convolutional neural networks. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14. pp. 467–482. Springer (2016)
27. Tan, J., Lu, X., Zhang, G., Yin, C., Li, Q.: Equalization loss v2: A new gradient balance approach for long-tailed object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1685–1694 (2021)
28. Tan, J., Wang, C., Li, B., Li, Q., Ouyang, W., Yin, C., Yan, J.: Equalization loss for long-tailed object recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11662–11671 (2020)
29. Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. International journal of computer vision **104**, 154–171 (2013)
30. Wang, C.Y., Bochkovskiy, A., Liao, H.Y.M.: Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7464–7475 (2023)
31. Wang, J., Zhang, W., Zang, Y., Cao, Y., Pang, J., Gong, T., Chen, K., Liu, Z., Loy, C.C., Lin, D.: Seesaw loss for long-tailed instance segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9695–9704 (2021)
32. Wang, T., Li, Y., Kang, B., Li, J., Liew, J., Tang, S., Hoi, S., Feng, J.: The devil is in classification: A simple framework for long-tail instance segmentation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16. pp. 728–744. Springer (2020)
33. Wu, J., Song, L., Wang, T., Zhang, Q., Yuan, J.: Forest r-cnn: Large-vocabulary long-tailed object detection and instance segmentation. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 1570–1578 (2020)
34. Zhang, C., Pan, T.Y., Chen, T., Zhong, J., Fu, W., Chao, W.L.: Learning with free object segments for long-tailed instance segmentation. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X. pp. 655–672. Springer (2022)
35. Zhang, S., Chen, C., Peng, S.: Reconciling object-level and global-level objectives for long-tail detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 18982–18992 (October 2023)