PRIORBOX: LONG-TAIL CALIBRATION WITH PRIORS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ABDULLAH DURSUN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

AUGUST 2022

Approval of the thesis:

PRIORBOX: LONG-TAIL CALIBRATION WITH PRIORS

submitted by **ABDULLAH DURSUN** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** ________________

Assist. Prof. Dr. Ramazan Gökberk Cinbiş
Supervisor, **Computer Engineering, METU** ________________

Examining Committee Members:

Assoc. Prof. Dr. Sinan Kalkan
Computer Engineering, METU ________________

Assist. Prof. Dr. Ramazan Gökberk Cinbiş
Computer Engineering, METU ________________

Prof. Dr. Selim Aksoy
Computer Engineering, Bilkent University ________________

Date: 31.08.2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: ABDULLAH DURSUN

Signature :

ABSTRACT

PRIORBOX: LONG-TAIL CALIBRATION WITH PRIORS

DURSUN, ABDULLAH
M.S., Department of Computer Engineering
Supervisor: Assist. Prof. Dr. Ramazan Gökberk Cinbiş

AUGUST 2022, 48 pages

Deep learning brought considerable improvements to computer vision, especially in recognition problems such as image classification, object detection, semantic segmentation, instance segmentation, and keypoint detection. These problems have critical applications in the real world, especially in the search, social media, and surveillance domains. Unfortunately, there is still a remarkable accuracy gap between research datasets and real-world deployments caused by data distribution disparity. In particular, most detection methods have a noticeable accuracy drop on datasets with long-tailed distributions due to the bias towards frequent classes.

This thesis describes PriorBox, which learns calibration factors for long-tail datasets utilizing class distributions and a simple convolutional neural network. Since PriorBox uses easy-to-collect distributional and spatial priors, it does not introduce any data collection steps. Furthermore, the proposed method does not include typical class-rebalancing and loss manipulation strategies and works well with the existing object detection and instance segmentation models. Simple distributional class priors, such as the number of instances, size and aspect ratio are shown to be helpful for improving detection results on rare classes without a significant impact on the inference speed. We thoroughly evaluate the approach on the LVIS dataset using the Mask R-CNN baseline on long-tail object detection and instance segmentation tasks.

Keywords: long-tail detection, long-tail segmentation, long-tail calibration, class imbalanced dataset

ÖZ

PRIORBOX: ÖN BİLGİ İLE DENGESİZ VERİ KALİBRASYONU

DURSUN, ABDULLAH
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Dr. Öğr. Üyesi. Ramazan Gökberk Cinbiş

Ağustos 2022 , 48 sayfa

Derin öğrenme görsel sınıflandırma, nesne tespiti, anlamsal bölütleme, nesne bölütleme ve kilit nokta tespiti gibi zorlu problemlere katkılarıyla bilgisayarlı görü alanına büyük yenilikler getirmiştir. Bu problemler arama motorları, sosyal medya ve askeri uygulamalarda önemli bir yer almaktadır. Maalesef günümüzde araştırma ve gündelik hayat uygulamalarındaki başarımlar arasında veri dağılım ayrımından kaynaklı ciddi farklar bulunmaktadır. Dengesiz veri setlerindeki çok örnekli sınıflara olan eğilimden dolayı çoğu nesne tespit yöntemi dikkate değer başarı düşüşüne uğramaktadır.

Bu tez dengesiz veri setleri için sınıf dağılımı ön bilgisi ile basit bir evrişimli sinir ağı kullanarak kalibrasyon çarpanı bulan PriorBox yöntemini önermektedir. PriorBox hazırlaması kolay dağılımsal ve uzamsal ön bilgileri kullandığı için ekstra veri toplama aşaması gerektirmemektedir. Önerilen yöntem yaygın sınıf dengeleme ve hata manipülasyon yöntemlerini kullanmayıp var olan nesne tespit ve bölütleme çalışmalarıyla uyumludur. Örnek sayısı, boyut ve en boy oranı gibi basit ön bilgiler az örnekli sınıfların tespitini çıkarım hızını değiştirmeden iyileştirebilmektedir. LVIS veri seti ve Mask R-CNN modeliyle dengesiz veri setlerinde nesne tespiti ve bölütleme problemleri üzerinde detaylı incelemeler yapılmıştır.

Anahtar Kelimeler: dengesiz veri setlerinde nesne tespiti, dengesiz veri setlerinde nesne bölütleme, dengesiz veri kalibrasyonu, dengesiz veri setleri

To the research community.

ACKNOWLEDGMENTS

I want to express my gratitude and respect to my supervisor Assist. Prof. Dr. Ramazan Gökberk Cinbiş for his continuous guidance. This thesis wouldn't exist without his motivating and fascinating supervision.

I am grateful to the thesis committee members, Assoc. Prof. Dr. Sinan Kalkan and Prof. Dr. Selim Aksoy, for their valuable time and comments.

Finally, I am thankful to my wife, Serpil, and my parents for their unconditional love, excitement, and support.

The work in this thesis was supported in part by the TUBITAK Grant 119E597. The numerical calculations reported in this thesis were partially performed at TUBITAK ULAKBIM, High Performance and Grid Computing Center (TRUBA resources).

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Networks
R-CNN	Regions with CNN Features
AP	Average Precision
AP_r	Average Precision Rare Classes
AP_c	Average Precision Common Classes
AP_f	Average Precision Frequent Classes
AP_s	Average Precision Small Instances
AP_m	Average Precision Medium Instances
AP_f	Average Precision Frequent Instances
mAP	Mean Average Precision
COCO	Microsoft COCO: Common Objects in Context
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
SVM	Support Vector Machines
IoU	Intersection over Union
RoI	Region of Interest
SPP	Spatial Pyramid Pooling
RPN	Region Proposal Network
NorCal	Normalized Calibration
CE	Cross-entropy
RFS	Repeat Factor Sampling
GT	Ground-truth
EQL	Equalization Loss
BAGS	Balanced Group Softmax
LVIS	Large Vocabulary Instance Segmentation

ACSL	Adaptive Class Suppression Loss
RS	Rank & Sort

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Definition

Object detection locates instances in an image or video that belong to the predefined categories. Detecting objects accurately and quickly allows building applications in domains like self-driving cars, surveillance, and search. Recent advancements in deep learning methods bring detectors with higher accuracies with faster predictions. However, these methods still have limitations, especially in learning categories with fewer samples.

Imbalanced datasets lead to learning disparity between frequent and rare classes. Since some categories are more common in the real world, deep learning models perform suboptimally detecting instances with limited samples. Following the taxonomy by Oksuz et al. [1], foreground-foreground class imbalance limits the generalization potential of the computer vision methods.

There are two prominent approaches to training detectors with long-tailed datasets. Re-sampling methods attempt to reduce the class imbalance effect by oversampling and under-sampling techniques [2, 3, 4]. However, there is a trade-off between different sampling procedures; oversampling causes overfitting for rare classes and under-sampling results in under-represented frequent classes. Re-weighting methods assign class and sample-level weights (generally higher weights to rare ones) in loss functions during training to overcome the class imbalance [5, 6, 7, 8]. Increasing weights for classes with fewer samples accelerates their learning in the networks. Unfortunately, these methods typically introduce data distribution-dependent hyperparameters to tune [6, 8].

1.2 Proposed Method

Since data distribution introduces a strong bias towards classes with a large number of samples, utilizing the prior information to calibrate the prediction scores improves the detection and segmentation tasks on both frequent and rare categories.

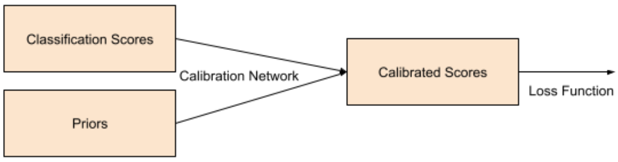


Figure 1.1: Proposed method.

The proposed method, PriorBox, works on the classification head of the detection networks and calibrates their outputs using easy-to-collect distributional and spatial priors and a simple 3-layer convolutional network without modifying the data sampling and loss calculation procedures (Figure 1.1).

This thesis work integrates image count, instance count, size count, and aspect ratio priors into the well-known Mask R-CNN [9] architecture without additional data collection and grouping efforts. Since the method doesn't depend on the specific architectures and loss functions, it could be seen as a general calibration framework.

1.3 The Outline of the Thesis

Chapter 1 introduces the problem, its importance, existing approaches, and the proposed method. Chapter 2 demonstrates the background information about object detection and alternative methods to the long-tail recognition problem. Chapter 3 explains how the prior box is constructed and how calibration factors are calculated and applied. Chapter 4 presents the ablation studies, their results, and comparisons with the existing methods. Finally, Chapter 5 summarizes PriorBox and potential future work.

CHAPTER 2

LITERATURE REVIEW

Convolutional neural networks (CNNs) are the building blocks of computer vision research. Their composable nature allows neural networks to learn hierarchical knowledge from low-level features to high-level concepts. They are commonly used in image classification, object detection, semantic segmentation, instance segmentation, and key-point detection tasks.

This chapter will briefly introduce the existing object detection and instance segmentation models to put the proposed model into historical context. Section 2.1 summarizes two-stage object detection models. Section 2.2 discusses recent work on long-tail recognition problems. Section 2.3 presents context-based rescoring methods.

2.1 Object Detection

Object detection models localize object instances of predefined classes in an image. After Krizhevsky et al. [10] achieved remarkable classification results in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [11], CNNs with deep neural architecture received noticeable attention among the object detection researchers.

In this thesis, we focus on *rescoring* the outputs from the classification branch of two-stage detection methods, especially R-CNN variants, as they are commonly adopted as baselines in long-tail recognition works.

2.1.1 R-CNN: Regions with CNN features

R-CNN [4] introduced by Girshick et al. is a multi-stage object detection model that provides a strong baseline for several state-of-the-art follow-up works. For each image, the R-CNN model creates approximately 2000 class-independent object proposals, computes fixed-length features from them, and classifies the features with class-specific linear support vector machines (SVMs).

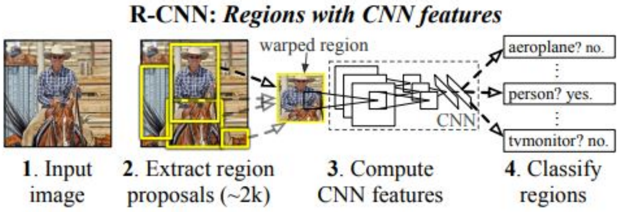


Figure 2.1: R-CNN model. Taken from R-CNN: Regions with CNN features [4].

R-CNN utilizes Selective Search [12] to generate object proposals and extracts 4096-dimensional features through pre-trained classification model[1]. After the class-specific SVMs compute the class scores, non-maximum suppression is applied for each class separately to eliminate the overlapping regions. Only regions with more than 0.3 intersection over union (IoU) overlap with ground truths considered positive examples.

Girshick et al. [4] discover that fine-tuning pre-trained classification models with detection datasets boost the mean average precision score. Also, utilizing larger classification CNN architectures, switching the 7-layer AlexNet [10] with 16-layer VGG [13] improved the mAP even more.

Furthermore, the authors attempted to apply bounding-box regression to object proposals from Selective Search. Integrating class-specific bounding-box regressors that exploit features extracted from the CNN provided another improvement.

[1] Ablation study contains different architectures pre-trained on ILSVRC2012 classification dataset [11].

2.1.2 Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition

SPP-net [14] published by He et al. brings significant speed improvements over R-CNN by computing the input image's feature map only once.

Even though CNNs can work on arbitrary-sized inputs (images or feature maps) and preserve their aspect ratio, the input sizes are manipulated to ensure fixed-sized outputs are passed to the fully-connected layers. In other words, the inputs are adjusted just to be consistent with the fully-connected layers.

He et al. [14] propose a spatial pyramid pooling (SPP) method to overcome the fixed-size input requirement imposed by the fully-connected layers. Since SPP converts the arbitrary-sized inputs into fixed-length vectors, deformations introduced by the operations like cropping and warping would be avoided. Placing the SPP layer between the convolutional and fully-connected layers eliminates the need for pre-processing the inputs.

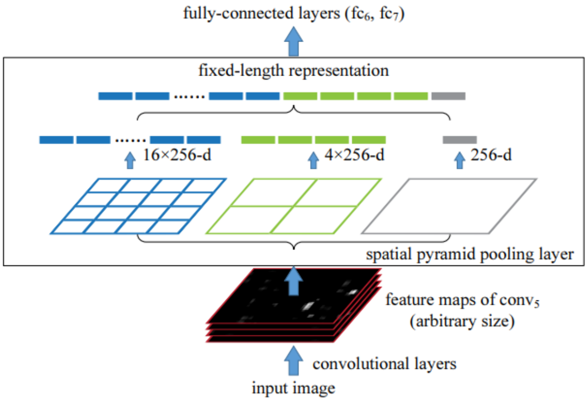

fully-connected layers (fc_6, fc_7)
fixed-length representation
16×256-d
4×256-d
256-d
spatial pyramid pooling layer
feature maps of $conv_5$
(arbitrary size)
convolutional layers
input image

Figure 2.2: Spatial Pyramid Pooling layer. Taken from SPP-net [14].

In theory, SPP would apply max-pooling to the channels of the feature map with the arbitrary number of bins (16, 4, and 1 in Figure 2.2) and concatenate the outputs to get a fixed-sized tensor without applying any transformations to the input. Since GPUs are optimized to work on fixed-sized inputs, for each epoch, the authors resized all training images to a fixed shape and performed the training. By using the same weights among the epochs, SPP-net simulates input shape invariant training. During inference, spatial pyramid pooling is applied to arbitrary image shapes without resizing.

Even though SPP-net [14] achieves comparable results to R-CNN [4], speed improvements and bringing scale-invariant training/testing to deep neural networks are remarkable contributions.

2.1.3 Fast R-CNN

Fast R-CNN [15], the second version of R-CNN [4], reduces training and inference durations and improves detection accuracies remarkably. It also utilizes object proposals from Selective Search [12].

Like SPP-net [14], Fast R-CNN computes the feature map of the input image once and extracts the region of interests (RoIs) on the feature map for each object proposal. Each region of interest is passed to the pooling layer to compute fixed-size features for each object proposal. The pooling layer divides the region of interest into *H x W* equal windows and applies max-pooling to the channels independently to extract a fixed-sized tensor, which is passed to the fully-connected layers to compute the RoI feature vector (Figure 2.3).

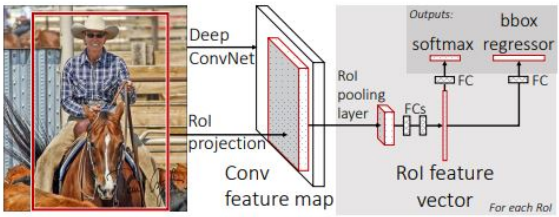


Figure 2.3: Fast R-CNN model. Taken from Fast R-CNN [15].

After one feature vector per object proposal is obtained, the network produces two outputs for the multi-task loss:

1. Class probabilities[2]
2. Per-class bounding-box offsets

[2] One extra class represents the catch-all background class.

Fast R-CNN calculates log loss from class probabilities and L_1 loss bounding box offsets and jointly optimizes their summation. During the optimization, the pre-trained VGG [13] layers are also trained in an end-to-end fashion.

2.1.4 Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

After the speed improvements from re-using image features among the object proposals, region proposal methods (like Selective Search [12]) become the bottleneck. Earlier work, [4], [14] and [15], utilized region proposals from fixed object proposal methods and tried fine-tuning them through regression offsets. Even though the bounding-box regression boosted the detection precision, it wasn't using deep CNNs' potential entirely. Depending on external proposal generation methods limits the learning capacity of the deep neural networks.

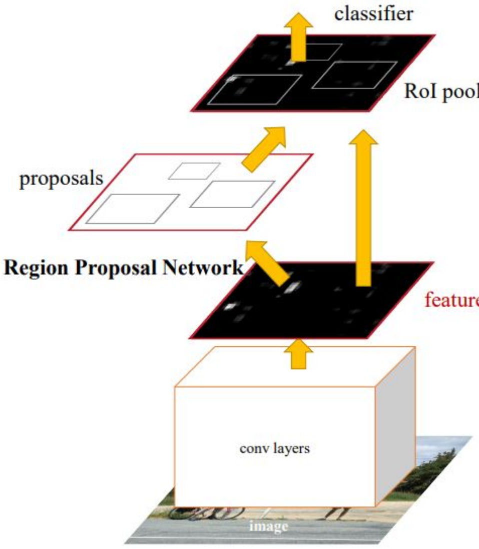


Figure 2.4: Shared convolution layers between Region Proposal Network and Fast R-CNN. Taken from Faster R-CNN [16].

Faster R-CNN [16] introduces a Region Proposal Network (RPN) that utilizes the same features as the detection network. RPN extracts region proposals with objectness scores and passes them to Fast R-CNN [15] detector without introducing a noticeable speed penalty (Figure 2.4).

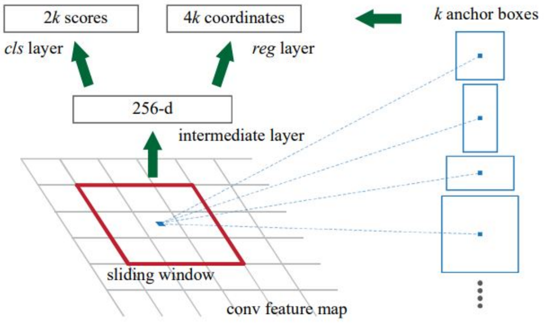


Figure 2.5: Sliding-window in RPN. Taken from Faster R-CNN [16].

RPN slides *n x n* spatial window[3] over the feature map extracted by the shared convolutional network. As shown in Figure 2.5, 9 proposals called *anchors* are generated with 3 scales and 3 aspect ratios for each sliding-window location. These multi-scale anchors allow Faster R-CNN to learn efficiently with single-scale inputs.

2.1.5 Mask R-CNN

Mask R-CNN [9] brings instance segmentation capabilities by adding a third branch to the Faster R-CNN [16] architecture. The mask branch predicts a pixel-to-pixel mask for each RoI extracted from RPN [16] proposals.

Since pixel-to-pixel instance segmentation requires higher precision than bounding-box-based object detection, He et al. introduced *RoiAlign* [9] to replace *RoiPool* [15]. RoiAlign eliminates the quantization of the RoI boundaries and bins by utilizing floating-number boundary coordinates and point locations as seen in Figure 2.6.

The mask branch predicts a separate mask for each class for each proposal by applying a per-pixel sigmoid. During training, the average cross-entropy loss is used. On the other hand, only the class predicted by the classification branch is used during the testing.

He et al. [9] show that Mask R-CNN outperforms state-of-the-art object detection and instance segmentation methods on the COCO 2016 challenge. Moreover, replacing

[3] *n = 3* is used in the implementation.

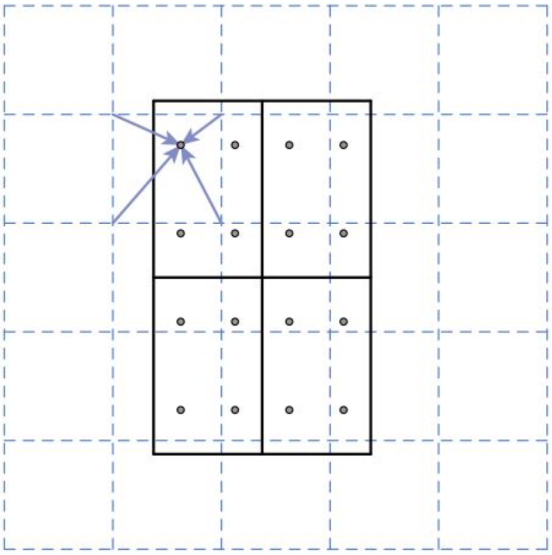


Figure 2.6: RoiAlign. Taken from Mask R-CNN [9]. Point values are calculated through bilinear interpolation from the closest grid points.

RoiPool with RoiAlign alone in Faster-RCNN improves detection precision.

2.1.6 Summary

To sum up, **R-CNN** [4] proposed fine-tuning pre-defined classification models, **SPP-net** [14] utilized region of interests on feature maps instead of object proposals, **Fast R-CNN** [15] defined region of interest pooling, **Faster R-CNN** [16] introduced Region Proposal Network and **Mask R-CNN** [9] suggested per-pixel masks. These models provided state-of-the-art results for object detection and built the backbone for more complex problems like long-tail recognition and keypoint detection.

2.2 Long-tail Recognition

Naturally, some object classes exist more in the real world. We have more cars than balloons, more cats than tigers, etc. Research datasets are influenced by this phenomenon. Considering deep neural networks are sensitive to training data distributions, they under-perform on samples from the rare classes.

An intuitive solution to the class imbalance problem is sampling rare classes more than the frequent classes (or sampling frequent classes less than the rare ones). Unfortunately, these solutions have the drawback of overfitting the rare classes and underfitting the frequent classes.

Another straightforward balancing idea is manipulating the network loss by assigning weights to classes and boosting learning from the rare classes. Unfortunately, this method brings new hyperparameters to tune.

2.2.1 LVIS: A Dataset for Large Vocabulary Instance Segmentation

Large Vocabulary Instance Segmentation (LVIS) [17] extracts a highly class imbalanced dataset from the COCO 2017 images. Having a large number of classes (1203 in LVIS v1) with imbalanced data distribution brings new challenges to object detection research.

Gupta et al. [17] propose a data resampling method named *repeat factor sampling* (RFS) that resamples images with respect to the largest repeat factor of the classes they contain. For each class c, repeat factor r_c is defined as:

$$r_c = max(1, \sqrt{t/f_c}), \tag{2.1}$$

where t is a hyperparameter that controls when the oversampling starts and f_c is the fraction of images that contains at least one instance of class c. Therefore, if a class exists in a quarter of the threshold images ($t = 0.001$ and $f_c = 0.00025$), those images would be sampled at least twice. RFS successfully improves detection performance on all class groups [17].

2.2.2 Class-Balanced Loss Based on Effective Number of Samples

Cui et al. [8] studied the data overlap problem in long-tailed distributions and proposed a loss re-weighting method based on the effective number of samples, defined as:

$$E_n = (1 - \beta^n)/(1 - \beta) \tag{2.2}$$

$$\beta = (N - 1)/N, \tag{2.3}$$

where n is the number of samples and N is the number of unique prototypes for a class. Since E_n is proportional to n, the authors scaled classification losses with $1/E_n$ to reduce losses from the head classes.

Applying the effective number of samples based class-balancing to common loss functions: Softmax CE loss, Sigmoid CE loss and Focal loss [18] on long-tailed versions of popular datasets: CIFAR-10 [19], CIFAR-100 [19], iNaturalist 2017-2018 [20] and ILSVRC 2012 [11] reduces the classification error rate [8].

2.2.3 Equalization Loss for Long-Tailed Object Recognition

Since frequent classes have more positive samples than others, negative gradients diminish learning rare classes. Equalization Loss (EQL) [6] ignores the discouraging gradients on the overwhelmed tail classes by introducing a loss weight term (w) for each region proposal. For foreground object proposals, losses from the non-ground-truth rare class are omitted. As a result, gradient norms of negative samples are shortened and predicted probabilities for the rare classes are boosted (Figure 2.7). Tan et al. [6] achieved first place in the LVIS Challenge 2019 by applying EQL. However, EQL requires tuning a dataset-dependent hyperparameter: frequency threshold (λ). PriorBox doesn't utilize manually-set model parameters; instead, it uses a trainable model conditioned on distributional and spatial priors.

2.2.4 Balanced Meta-Softmax for Long-Tailed Visual Recognition

Object recognition metrics like accuracy and mean average precision don't consider imbalanced data distributions. Ren et al. [3] defines ϕ and $\hat{\phi}$ as conditional distribution on the balanced test set and imbalanced training set respectively and try to minimize the disparity between their posteriors by modifying the loss function as

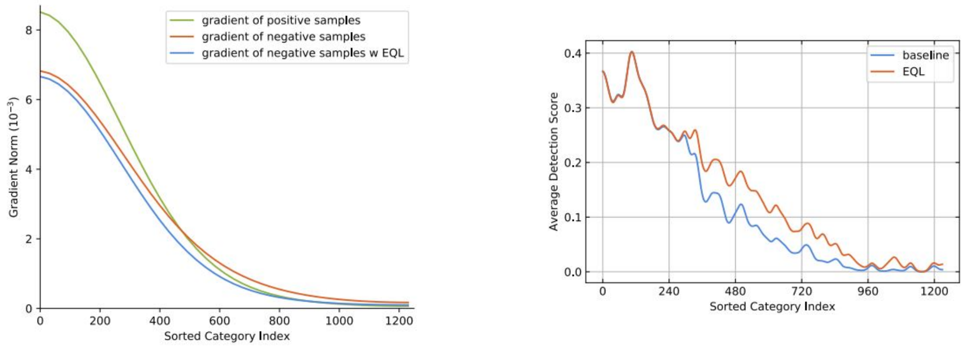


Figure 2.7: Gradient norms and average detection probabilities. Taken from EQL [6].

follows:

$$-\log(\hat{\phi_y}) = -\log\left(\frac{n_y e^{\eta_y}}{\sum_{i=1}^{k} n_i e^{\eta_i}}\right), \tag{2.4}$$

where n_c is the number of samples for class c, η_c is the model output for class c and k is the number of classes. Balanced Softmax (Eq. 2.4) improves both classification and detection results by integrating the number of instances into the loss function.

2.2.5 Overcoming Classifier Imbalance for Long-tail Object Detection with Balanced Group Softmax

Li et al. [21] investigated the classification head's effect on long-tail object detection. They analyzed the relation between the number of instances per class and classification head weight norms of the pre-trained Faster R-CNN model on LVIS and COCO (Figure 2.8).

While the difference between class weight norms is significant for LVIS, COCO's weight norms are more balanced. Since the LVIS dataset has a higher class imbalance and the head classes have more samples than the rare ones, their weights are activated more and rare class weights are suppressed.

The authors proposed Balanced Group Softmax (BAGS) [21] that computes softmax and cross-entropy loss in disjoint groups created with respect to the number of instances. As weights of the classes with similar distribution are trained in isolation, weights from the rare classes find more chance for updates.

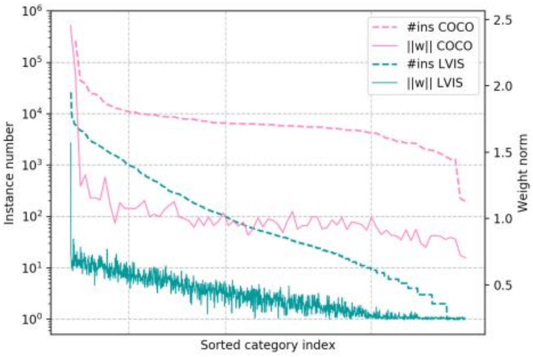


Figure 2.8: Classifier weight norms of Faster R-CNN on LVIS and COCO. Retrieved from [21].

In practice, the authors created four groups for the foreground classes and 1 for the background and improved RFS [17] baseline. Yet, their method is susceptible to group boundaries. Wang et al. investigated different boundaries and AP_r results reduced drastically [22].

2.2.6 Adaptive Class Suppression Loss for Long-Tail Object Detection

Wang et al. analyzed the shortcomings of the group-based long-tail recognition methods (for example, Equalization Loss [6] and Balanced Group Softmax [21]). Since the group boundaries play a critical role in the detection performance, the generalization power of these models on datasets with various data distributions is low. Moreover, grouping classes with respect to the frequencies leads to training semantically irrelevant examples similarly.

As rare classes are suppressed severely by the negative gradients from the head classes, Adaptive Class Suppression Loss (ACSL) [22] skips loss terms from unrelated categories determined by the probability scores. As a result, rare class weights receive fewer updates while learning frequent classes. ACSL improves both baseline [17] and group-based [6, 21] models on LVIS v0.5 [17] without utilizing any priors like frequency and size.

2.2.7 The Devil is in Classification: A Simple Framework for Long-tail Object Detection and Instance Segmentation

SimCal [23], which won the 2019 LVIS challenge, reveals that the performance drop in long-tail object detection and instance segmentation models is mostly caused by the proposal classification phase.

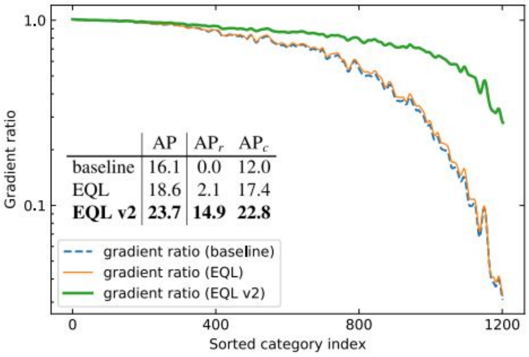
To demonstrate the classification bias toward frequent classes, the authors set the classification results of the proposals from RPN [16] as the ground-truth values. As a result, the detection and segmentation AP scores improved drastically on LVIS v0.5.

The authors propose a decoupled learning architecture to minimize the classification bias toward frequent classes. They freeze all layers except the classification head of a pre-trained Mask R-CNN model and fine-tune the classifier with balanced proposals sampled from the original dataset. Even though SimCal achieved better results on rare classes, the performance on head classes dropped.

2.2.8 Equalization Loss v2: A New Gradient Balance Approach for Long-tailed Object Detection

After analyzing the gradient norms in EQL [6], Tan et al. studied the gradient ratio of positives to negatives and published Equalization Loss v2 [7]. They noticed that frequent and rare classes exhibit different behaviors. Even though the magnitudes for positive and negative gradients for head classes were similar, negative gradients dominated the positives in tail classes (Figure 2.9).

EQL v2 [7] is a re-weighing mechanism that utilizes the gradient ratio to minimize the effects of overwhelming negative gradients. The method balances negative and positive gradients especially on rare classes (Figure 2.9) and achieves remarkable results on long-tail detection tasks.



	AP	AP_r	AP_c
baseline	16.1	0.0	12.0
EQL	18.6	2.1	17.4
EQL v2	**23.7**	**14.9**	**22.8**

Figure 2.9: Gradient ratios of classification loss. Retrieved from [7].

2.2.9 Rank & Sort Loss for Object Detection and Instance Segmentation

Oksuz et al. [24] propose a ranking-based loss function that sorts both positives above negatives and positives between themselves. Furthermore, the proposed loss function, Rank & Sort (RS) loss, optimizes the average precision metric directly. Even though RS loss achieved remarkable results in long-tail visual detection tasks, we adopt cross-entropy loss to be compatible with more models.

2.2.10 From Generalized Zero-shot Learning to Long-tail with Class Descriptors

Samuel et al. [25] integrate class descriptions to balance the bias toward frequent classes. Information from class descriptions allows DRAGON [25] to de-bias predictions on a sample-by-sample basis and boost tail class classification accuracy.

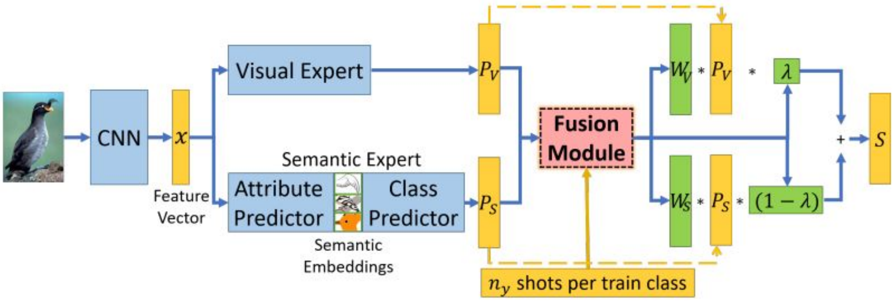


Figure 2.10: The DRAGON architecture. Retrieved from [25].

As semantic class descriptions are effective in few-shot and zero-shot recognition tasks [26, 27, 28], the authors designed a semantic expert network to tune CNN outputs with a residual block (Figure 2.10). Visual expert, which is a conventional CNN, predicts head classes successfully and semantic expert improves the accuracy of the tail classes. Furthermore, *fusion module* reweighs prediction scores with a polynomial regression that utilizes the number of samples for each class. The authors also generated long-tail variants of the existing datasets [29, 30, 26] and shared extensive experiments with them.

2.2.11 Evaluating Large-Vocabulary Object Detectors: The Devil is in the Details

Dave et al. [31] assessed average precision metric on long-tailed LVIS [17] dataset and published how it can be improved easily. Even though AP is calculated in a class-independent approach, limiting the number of detections in an image causes ranking across classes.

The authors shared that increasing the number of detections per image alone boosts the AP remarkably on the LVIS dataset. On the other hand, it doesn't have the same effect on the COCO dataset with 80 classes. This behavior suggests that the predictions for the rare classes might be ignored due to the detection limit. Since LVIS has 1203 classes, most classes won't affect the evaluation with the 300 detection limit.

Dave et al. [31] defined AP^{Fixed} as leaving the number of detections unlimited per image, but limited per class. For example, the LVIS validation and test sets have 20000 images, they suggested using 10000 detections per class. Moreover, the recent methods on long-tail recognition (EQL [6], BAGS [21], and Federated loss [32]) don't improve AP^{Fixed} as they advance AP.

2.2.12 On Model Calibration for Long-Tailed Object Detection and Instance Segmentation

Normalized Calibration (NorCal) [33] reweighs class scores to reduce the bias toward frequent classes by utilizing the number of samples per class. NorCal does not require re-training or fine-tuning of models; it just calibrates and normalizes pre-trained model outputs during testing phase to improve detection success.

NorCal [33] scales the class outputs with respect to the number of images that contain at least one instance of class c (denoted by N_c). One exception is the background class. Since the background class exists in large amounts in all examples, it requires a special approach.

The authors decompose softmax function by multiplying both nominator and denominator by $\sum_{c'=1}^{C} \exp\left(\phi_{c'}(x)\right)$:

$$p(c|x) = \frac{\sum_{c'=1}^{C} \exp\left(\phi_{c'}(x)\right)}{\sum_{c'=1}^{C} \exp\left(\phi_{c'}(x)\right) + \exp\left(\phi_{C+1}(x)\right)} \times \frac{\exp\left(\phi_{c}(x)\right)}{\sum_{c'=1}^{C} \exp\left(\phi_{c'}(x)\right)} \tag{2.5}$$

For each proposal, the first term on the right-hand side stands for the possibility of a foreground class and the second term represents the possibility of class c.

Since the background class only exists in the first term, calibrating its score wouldn't affect other class scores. Therefore, background class scores are normalized without calibration.

Intuitively the scores should be penalized more with the increasing sample count per class. NorCal scales down the exponential of the outputs $\phi_c(x)$ by the power of N_c as follows:

$$p(c|x) = \frac{\exp\left(\phi_{c}(x)\right)/a_c}{\sum_{c'=1}^{C} \exp\left(\phi_{c'}(x)\right)/a_c + \exp\left(\phi_{C+1}(x)\right)}, \tag{2.6}$$

where

$$a_c = N_c^{\gamma}, \gamma \geq 0. \tag{2.7}$$

To sum up, NorCal calibrates the proposal class scores by a monotonically increasing term with a single hyperparameter $\gamma = 0.6$. It improved LVIS v1 [17] baseline model RFS[4] in both object detection and instance segmentation, especially on rare classes.

2.3 Context-based Rescoring

The rescoring mechanisms have also been proposed to improve detections through contextual priors, e.g., Cinbis et al. [35] and Pato et al [36]. These methods typically take initial detections and revise the detection scores to find a contextually consistent set of detections. Our proposed approach is also similar in the spirit of learning to correct initial detections. However, instead of contextual priors, we focus on easy-to-collect distributional priors to handle class imbalances.

[4] Mask R-CNN [9] model with ResNet-50 [34] backbone and repeat factor sampling (RFS) [17].

CHAPTER 3

METHOD

This chapter explains how PriorBox calibrates the classification scores by modifying the classification head of object detection and instance segmentation architectures. It first describes why data imbalance is a problem and defines how the output of the loss function is combined with the priors for calibration.

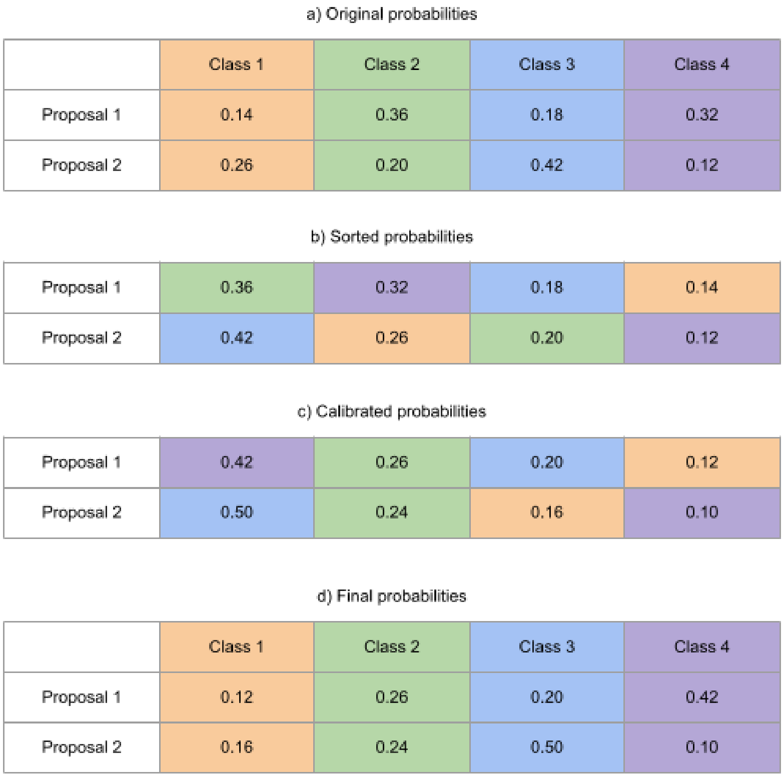
3.1 Problem

Modern two-stage object detection and instance segmentation models optimize multiple targets, classification, box regression, and mask prediction. As investigated in [23, 8, 21, 22], the main reason behind the inferior results in long-tail detection is the classification head. Since rare classes have only a few samples, they are suppressed by the other classes with much more positive gradients. Therefore, the networks tend to expand the scores of the frequent categories. Considering the softmax layer used to calculate the class probabilities, the foreground-foreground imbalance problem becomes more critical with the increased number of classes, like 1203 categories in LVIS v1.

3.2 Proposed Method

Object detection models predict class probability distributions for each region proposal, $p(c|x_i)$, where x_i is the feature representation for bounding box i and c is one of the classes from the dataset.

PriorBox improves object detection and instance segmentation results by injecting per-class distributional and spatial priors from the training data. In other words, the proposed method replaces the prediction target with $p(c|x_i, t_1, t_2, ...)$, where t_k represents prior k. Adding prior evidence to the probability distribution enhances both recognition tasks, especially on rare classes.



a) Original probabilities

	🟧 Class 1	🟩 Class 2	🟦 Class 3	🟪 Class 4
Proposal 1	0.14	0.36	0.18	0.32
Proposal 2	0.26	0.20	0.42	0.12

b) Sorted probabilities

Proposal 1	🟩 0.36	🟪 0.32	🟦 0.18	🟧 0.14
Proposal 2	🟦 0.42	🟧 0.26	🟩 0.20	🟪 0.12

c) Calibrated probabilities

Proposal 1	🟪 0.42	🟩 0.26	🟦 0.20	🟧 0.12
Proposal 2	🟦 0.50	🟩 0.24	🟧 0.16	🟪 0.10

d) Final probabilities

	🟧 Class 1	🟩 Class 2	🟦 Class 3	🟪 Class 4
Proposal 1	0.12	0.26	0.20	0.42
Proposal 2	0.16	0.24	0.50	0.10

Figure 3.1: Sorting and calibration process, priors are omitted for brevity.

PriorBox is designed as a post-calibration method with a fine-tuning phase. After the base network is trained without the priors, PriorBox injects priors to the classification branch and calibrates the outputs. The modified network trains with the pre-trained weights, additional priors, and a lower learning rate.

Since PriorBox doesn't modify the output of the classification branch (Figure 3.1), it could be integrated into various detection architectures.

3.3 Priors

Priors are the building blocks of the prior box. They provide class and proposal-wise external information to the classification branch. We focus on common, easy-to-collect priors to develop a low-effort and highly generalizable calibration method. All priors are extracted from the LVIS v1.0 [17] training split using 20 lines of code scripts. Task-specific priors might improve the results even more on more specialized problems.

This section will explain the priors placed in the prior box and how they could be suitable for calibration.

3.3.1 Instance and Image Counts

Instance count prior denotes the number of ground truth bounding boxes in the training split for each class. Likewise, image count is the number of images a class has at least one instance. Fortunately, the LVIS class metadata contains *instance_count* and *image_count* fields.

Instance and image counts are commonly utilized in existing long-tail and few-shot learning methods [8, 6, 3, 21, 25, 33]. Intuitively, we expect to increase scores and probabilities of rare classes more than the frequent ones. The convolutional network compares the original class scores considering the count priors and calibrates them accordingly.

Both priors are normalized separately by dividing their maximum, so the prior box values are between 0 and 1. As proposal properties don't change instance and image counts, we utilize them as fixed, class-wise priors.

3.3.2 Size Counts

The COCO [37] evaluation metric splits object instances into three groups: small, medium and large, considering their bounding-box areas. Images smaller than $32x32$ are small, between $32x32$ and $96x96$ are medium and others are large.

Similarly, for each ground truth box in the LVIS training split, we count each class and size and build a 1203×3 vector. Furthermore, the counts are normalized by dividing the group-wise maximum count.

Since each proposal has a different size, the prior box contains dynamic values for the size count prior. For example, for a medium-sized proposal, the corresponding row in the prior box contains the normalized counts for the medium group. Thus, while calibrating the scores, the convolutional network boosts the class scores of proposals with respect to the size count relevance.

3.3.3 Aspect Ratio

Aspect ratio is another easy-to-collect prior PriorBox utilizes. For each proposal, the absolute difference between the average class aspect ratio and proposal aspect ratio is placed inside the prior box. The motivation is to inject aspect ratio similarity information into the calibration network.

3.4 Prior Box

The DRAGON architecture [25] reorders both expert outputs with respect to the decreasing prediction scores to make the convolution meaningful along the class scores axis. Ordering and applying convolutions between the class outputs allow evaluating and calibrating scores regarding the priors. Considering the number of instances and average aspect ratio priors, if the probabilities of two classes are the same, the class with the lower number of samples and less aspect ratio difference would have more potential to be the ground truth.

Assuming we have 1000 proposals per image in a 1200 class detection task with three priors, the prior box would have the shape of **(1000, 1200, 4)** [1] as in Figure 3.2. For class-wise priors, the prior row is expanded by the number of proposals.

Similar to DRAGON [25], the priors in the prior box are ranked with respect to the classification scores from the original network (Figure 3.1). After the original prob-

[1] Additional dimension contains the original scores.

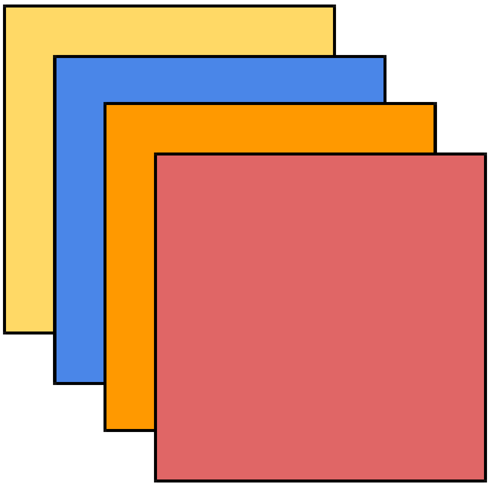


Figure 3.2: Prior box with three priors. The first two-dimensional vector represents the original scores, others are priors expanded by the number of proposals.

abilities are extracted from the network, each proposal's probabilities and priors are sorted separately. Then, the sorted tensor is passed to the calibration network, a simple 3-layer convolutional network, to update the probabilities. Finally, the calibrated probabilities return to the original indices to be compatible with the network's loss function.

3.5 Calibration Network

After building the prior box, a convolutional network is applied to compute the calibration factor. This calibration factor mutates the network logits and probabilities to debias the data imbalance effects.

As the instance proposals are independent in an image, applying a convolution along the proposals axis would not be effective. Instead, the calibration factors are learned from the score differences and priors by traveling on the classes axis.

The first convolutional layer has one input channel for each prior and the last one has a single output channel (Figure 3.3). The convolutional network preserves the original network's output shape by having a stride of 1 and adding padding as needed. As the

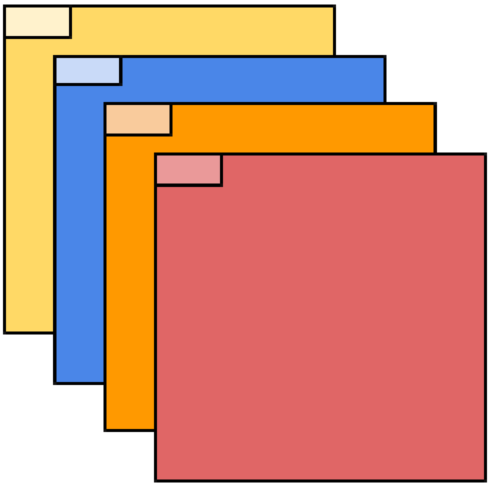


Figure 3.3: Channels for the first convolutional layer on the prior box.

prior box elements are sorted by the class scores, an unsort operation is applied before the loss calculation. PriorBox does not require any changes to the loss functions.

As rare classes have a few samples, the calibration network is a simple 3-layer convolutional network. Complex networks require more examples to learn.

To sum up, we use the following process to learn calibration factors for long-tail object detection:

1. Pass the input image to the original network (for example, Mask R-CNN) and extract the classification head scores (or classification probabilities)

2. Build the prior box, which is sorted by the classification head scores (or classification probabilities)

3. Apply the convolutional network to get the calibration factors

4. Apply the calibration factors (directly output calibration network results or add them to the original scores/probabilities)

5. Unsort the results to the original indices

6. Calculate the loss with the original loss function

In other words, PriorBox recalculates the network outputs with the original scores, class, and instance-wise priors.

3.6 Handling the Background

Since the background class has an entirely different meaning and representation than the foreground classes, the calibration network doesn't update its scores and probabilities. After the calibration process, the background class probability is updated due to the normalization.

The ablation study results and implementation details are demonstrated in Chapter 4.

3.7 Calibration on Logits

Logits are the raw network outputs that are the inputs of the softmax layer. Calibrating the logits successfully would improve the probability distribution from the softmax layer.

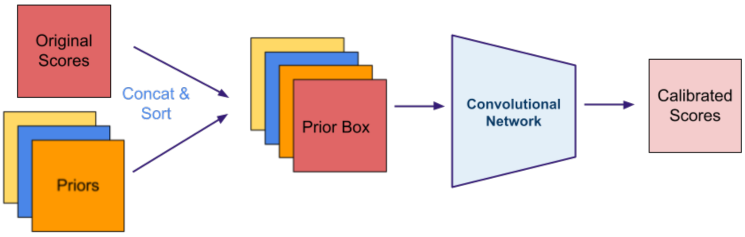


Figure 3.4: Calibration on logits.

There are two main strategies to employ the calibration factors extracted from the convolutional network:

1. Use calibration factors directly as the class scores,
2. Add calibration factors to the network's original class scores in a residual way.

3.8 Calibration on Probabilities

Since PriorBox is independent of the output type, class probabilities can be employed in the calibration network. Similar to the logit calibration, the prior box can be constructed by the sorted class probabilities.

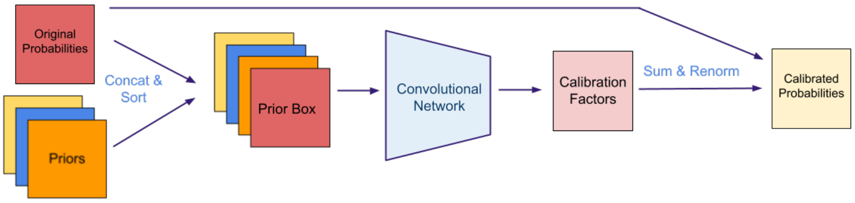


Figure 3.5: Calibration on probabilities.

CHAPTER 4

EXPERIMENTS

This chapter starts with the dataset and PriorBox implementation details. After that, we describe the issues on reproducibility and how we choose a strong baseline. Finally, we outline detailed experiments to evaluate the effectiveness of PriorBox with various priors and backbone networks.

4.1 Dataset

LVIS [17] is a recent, long-tail dataset built with COCO 2017 images [37] and brand-new instance segmentation and bounding box annotations. As it has a significantly larger number of classes (1203 classes in LVIS v1) than the COCO dataset (80 classes), annotating all instances with disjoint categories is impractical, so multiple ground truth values are acceptable. For hierarchical relations, instances are labeled for the most specific class and more general categories. Category names are selected from WordNet [38] synonyms and definitions.

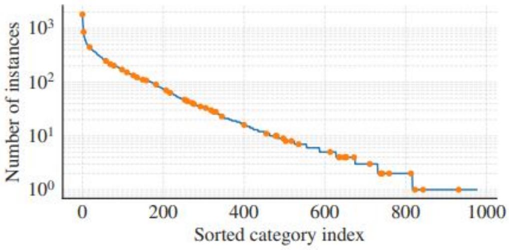


Figure 4.1: Class imbalance in LVIS, sampled from 5000 images. Retrieved from [17].

LVIS has a high foreground-foreground class imbalance, so learning representations for tail classes is critical to achieving valuable results. Figure 4.1 shows that head classes have thousands of samples while tail classes have just a few (for example, tennis racket appears in 1977 images, wardrobe appears in only 1).

Furthermore, LVIS categories are split into three groups considering the number of images they appear in: rare (1-10 images), common (11-100 images) and frequent (more than 100 images).

The following experiments utilize LVIS v1.0 training set (100170 images) for training and validation set (19809 images) for evaluation.

4.2 Evaluation Metrics

Average precision (AP), defined as the area under the precision-recall curve, is commonly used to compare different object detection methods. Microsoft COCO: Common Objects in Context [37] detection evaluation methods[1] are widely used in object detection research. AP50 and AP75 are calculated through predictions with more than 0.5 and 0.75 intersection over union ratios between proposal boxes and ground truths. Moreover, the primary challenge metric AP is defined as the average of average precisions between 0.5 and 0.95 with 0.05 steps.

LVIS [17] adopts COCO metrics with additional ones: AP_r for rare, AP_c for common and AP_f for frequent classes, as defined in Section 4.1.

4.3 Implementation Details

PriorBox utilizes the standard layers and models from PyTorch [39] and Detectron2 [40]. Mask R-CNN [9] model [2] with FPN [41] is adopted as the baseline model with pre-trained weights from [42].

During training, input images are resized to have at most 1333 pixels in the long edge

[1] https://cocodataset.org/#detection-eval

[2] https://github.com/facebookresearch/detectron2/blob/main/configs/LVISv1-InstanceSegmentation/mask_rcnn_R_50_FPN_1x.yaml

while keeping the aspect ratio unchanged. In addition, a horizontal flip is applied with 0.5 probability. During testing, input images are not resized.

All experiments, unless otherwise stated, are based on stochastic gradient descent (SGD) with an initial learning rate of 0.001, with a momentum of 0.9, batch size of 16 and 300 detections per image in a single GPU training for 14 epochs. Repeat Factor Sampling (RFS) [17] with a 0.001 repeat threshold is utilized in the data loaders.

To analyze the efficiency of PriorBox, we re-trained the baseline models with the same schedule for all calibration methods and compared the scores accordingly (*R50-FPN re-train* in Table 4.2, 4.3 and *R101-FPN re-train* in 4.4). Therefore, we do not attribute the improved results due to the longer training schedule to PriorBox.

4.4 Baseline

Reproducibility in deep learning research is a significant problem due to non-shared code, fast-changing software tools, sensitivity to implementation details, and sometimes secrecy. Finding a reproducible baseline is a challenging task.

Even though modern deep learning tools of interest (in particular PyTorch [39], TensorFlow [43], MMDetection [44] and Detectron2 [40]) have stabilized in recent years, there are sometimes significant differences among the pre-trained models belonging to the same works. For example, Detectron2's Mask R-CNN [9] model[3] outperforms MMDetection implementation[4] by **0.5** AP score. Furthermore, pre-trained (baseline) models might improve with more training, so comparing new methods (generally with longer training schedules) with them would be erroneous. For instance, MMDetection's LVIS baseline[5] scores improve remarkably with the increased number of epochs.

We utilize pre-trained weights for Mask R-CNN on LVIS v1 from MosaicOS [42] (*R50-FPN baseline* in Table 4.2, 4.3 and *R101-FPN baseline* in Table 4.4), which is

[3] https://github.com/facebookresearch/detectron2/blob/main/MODEL_ZOO.md#coco-instance-segmentation-baselines-with-mask-r-cnn

[4] https://github.com/open-mmlab/mmdetection/tree/master/configs/mask_rcnn#results-and-models

[5] https://github.com/open-mmlab/mmdetection/tree/master/configs/lvis#results-and-models-of-lvis-v1

Table 4.1: Mask R-CNN on LVIS v1 baseline from MosaicOS [42]

	Detection				Segmentation			
Backbone	AP	AP_r	AP_c	AP_f	AP	AP_r	AP_c	AP_f
R50-FPN [34]	23.3	12.3	21.3	30.2	22.6	12.3	21.3	28.6
R101-FPN [34]	25.4	14.7	23.6	32.3	24.8	15.2	23.7	30.3

commonly used in the latest studies [33, 45, 46].

4.5 Results

In this section, we share empirical results of various calibration approaches.

4.5.1 Calibration on Logits

Similar to ACSL [22], network outputs might indicate the learning status and they could be exploited for calibration. Building the prior box with the network logits has the potential to balance the scores and improve detection accuracy. Unfortunately, the network performed almost identical to the re-trained baseline model (*PriorBox no-priors* in Table 4.2). Without any relation to the external information, network scores without any priors aren't effective.

Frequent classes have higher logits due to their intensive positive gradients. Class scores and image counts could be helpful for calibration, so we built a prior box with them and applied the convolutional network. We expect PriorBox to boost scores from rare classes while reducing the frequent ones. PriorBox enhances both segmentation and detection baselines with the image count prior, especially on rare classes (**2.1** AP_r points over the baseline and **0.7** AP_r points over the re-trained baseline on instance segmentation, *PriorBox img* in Table 4.2).

Even though the number of instances is parallel to the image count, the more priors, the better. Thus, a new prior box is created with the scores, image count and instance count. Adding the number of instances to the prior box enhances AP_r results by

Table 4.2: **PriorBox calibration on logits** with ResNet-50 [34] backbone on LVIS v1. img denotes image count prior, ins denotes instance count prior, $size$ denotes size count prior, ar denotes aspect ratio difference prior, (R) denotes residual calibration.

	Detection				Segmentation			
Model	AP	AP_r	AP_c	AP_f	AP	AP_r	AP_c	AP_f
R50-FPN baseline	23.3	12.3	21.3	30.2	22.6	12.3	21.3	**28.6**
R50-FPN re-train	23.6	13.4	21.6	**30.3**	22.9	13.7	21.5	**28.6**
PriorBox no-priors	23.6	13.2	21.6	**30.3**	22.9	13.7	21.5	**28.6**
PriorBox no-priors (R)	23.6	13.6	21.7	**30.3**	22.9	13.7	21.5	28.5
PriorBox img	**23.8**	14.1	21.7	**30.3**	**23.1**	14.4	21.6	**28.6**
PriorBox img (R)	23.7	13.6	21.7	**30.3**	**23.1**	13.7	**21.8**	**28.6**
PriorBox img + ins	**23.8**	14.0	21.8	**30.3**	**23.1**	**14.6**	21.7	**28.6**
PriorBox img + ins (R)	23.6	13.1	21.7	**30.3**	23.0	13.9	21.6	**28.6**
PriorBox img + ins + $size$	**23.8**	14.1	21.7	**30.3**	**23.1**	14.4	21.6	**28.6**
PriorBox img + ins + $size$ (R)	23.6	13.7	21.6	30.2	23.0	14.0	21.6	**28.6**
PriorBox img + ins + $size$ + ar	**23.8**	13.9	**21.9**	**30.3**	**23.1**	14.1	**21.8**	**28.6**
PriorBox img + ins + $size$ + ar (R)	23.6	**14.2**	21.4	30.2	23.0	14.1	21.5	**28.6**

0.9 and **0.6** absolute improvement over the re-trained model on segmentation and detection tasks, respectively (*PriorBox img + ins* in Table 4.2).

Unlike the class-wise priors (image and instance counts), the size count and aspect ratio priors are dynamically calculated concerning the spatial properties of the proposal boxes. If a bounding box is small, the corresponding box in the prior box is adjusted accordingly. Even though the resulting models perform better than the baseline and the re-trained baseline, they don't bring considerable improvements over other PriorBox models (*PriorBox img + ins + size* and *PriorBox img + ins + size + ar* in Table 4.2).

Moreover, we trained residual logit calibration models (denoted with (R) in Table 4.2) and their results were better than the re-trained baseline but inferior to the non-residual counterparts.

Table 4.3: **PriorBox calibration on probabilities** with ResNet-50 [34] backbone on LVIS v1. img denotes image count prior, ins denotes instance count prior, $size$ denotes size count prior, ar denotes aspect ratio difference prior, (R) denotes residual calibration.

	Detection				Segmentation			
Model	AP	AP_r	AP_c	AP_f	AP	AP_r	AP_c	AP_f
R50-FPN baseline	23.3	12.3	21.3	30.2	22.6	12.3	21.3	**28.6**
R50-FPN re-train	**23.6**	13.4	**21.6**	**30.3**	22.9	13.7	21.5	**28.6**
PriorBox no-priors (R)	**23.6**	13.9	21.5	30.2	**23.0**	**14.3**	21.5	28.5
PriorBox img (R)	**23.6**	13.6	**21.6**	**30.3**	22.9	13.6	**21.6**	**28.6**
PriorBox img + ins (R)	**23.6**	14.0	21.5	30.2	**23.0**	14.1	21.5	28.5
PriorBox img + ins + $size$ (R)	**23.6**	13.7	**21.6**	30.2	**23.0**	14.0	**21.6**	**28.6**
PriorBox img + ins + $size$ + ar (R)	**23.6**	**14.2**	21.4	30.2	**23.0**	14.1	21.5	**28.6**

4.5.2 Calibration on Probabilities

Empirical results show that adding calibration factors to the original probabilities exhibits more stable training than predicting the class probabilities directly. After adding the calibration factors to the original probabilities, the results are L1 normalized before outputting the calibrated probabilities.

Similar to Section 4.5.1, PriorBox improves object detection and instance segmentation results on rare classes over the re-trained baseline models, **0.8** and **0.4** points, respectively (*PriorBox img + ins + size + ar (R)* in Table 4.3).

One notable result is that calibration over probabilities without priors works better than the logits-only method, meaning that probabilities are stronger signals than logits in representing the learning status.

4.5.3 Generalization to Larger Models

Training logit and probability calibration networks with ResNet-50 [34] backbone confirms that PriorBox advances the object detection and instance segmentation results. Still, to ensure the proposed method's generalization potential, we replace the

Table 4.4: **PriorBox calibration on logits and probabilities** with ResNet-101 [34] backbone on LVIS v1. img denotes image count prior, ins denotes instance count prior, $size$ denotes size count prior, ar denotes aspect ratio difference prior, L denotes calibration on logits, (PR) denotes residual calibration on probabilities.

	Detection				Segmentation			
Model	AP	AP_r	AP_c	AP_f	AP	AP_r	AP_c	AP_f
R101-FPN baseline	25.4	14.7	23.6	32.3	24.8	15.2	23.7	30.3
R101-FPN re-train	25.7	15.4	23.9	32.3	25.0	15.6	**24.0**	30.3
PriorBox img (L)	25.9	15.6	24.1	32.3	25.0	15.7	**24.0**	30.3
PriorBox img + ins (L)	**26.0**	**16.2**	24.2	**32.4**	25.1	**16.2**	**24.0**	30.3
PriorBox img + ins + $size$ (L)	25.8	15.8	23.9	**32.4**	24.9	15.6	23.8	30.3
PriorBox img + ins + $size$ + ar (L)	**26.0**	16.1	**24.3**	32.3	25.0	15.8	23.9	30.3
PriorBox img (PR)	25.9	**16.2**	24.0	32.3	**25.2**	**16.2**	**24.0**	**30.4**
PriorBox img + ins (PR)	25.7	15.8	23.8	32.3	25.0	**16.2**	23.8	30.3
PriorBox img + ins + $size$ (PR)	25.8	16.0	23.9	32.3	25.1	**16.2**	23.9	30.3
PriorBox img + ins + $size$ + ar (PR)	25.8	16.1	23.8	32.3	25.0	16.1	23.9	30.3

ResNet-50 backbone with its larger version, ResNet-101 [34]. Similar to the earlier models, PriorBox models boost the recognition performances (Table 4.4).

4.5.4 Generalization to Other Datasets

Even though COCO [37] has fewer classes with lower imbalance than LVIS [17], it still is one of the commonly used object detection and segmentation datasets. To evaluate PriorBox on the COCO dataset, we apply logit calibration with image and instance count priors to Detectron2's Mask R-CNN baseline[6]. PriorBox improves both object detection and instance segmentation AP results by **0.5** and **0.4** points, respectively. PriorBox performs better than or equal to the baseline for **77%** of the categories (Table 4.5 and Table 4.6).

[6] https://github.com/facebookresearch/detectron2/blob/main/configs/COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_1x.yaml

Table 4.5: Bounding box AP results for calibration with image and instance counts on COCO.

Category	Baseline	PriorBox	Category	Baseline	PriorBox	Category	Baseline	PriorBox
airplane	62.3	**62.7**	apple	18.8	**19.1**	backpack	14.0	**14.5**
banana	**21.6**	21.0	baseball bat	24.4	**25.3**	baseball glove	33.7	**33.8**
bear	64.2	**65.5**	bed	38.3	**38.5**	bench	21.3	**21.9**
bicycle	28.7	**29.2**	bird	35.3	**35.8**	boat	25.6	**26.2**
book	**14.1**	14.0	bottle	38.1	**38.4**	bowl	40.8	**40.9**
broccoli	23.6	23.6	bus	61.6	**62.4**	cake	33.1	**33.6**
car	42.8	**43.0**	carrot	20.7	**21.2**	cat	**62.5**	61.9
cell phone	33.6	**33.9**	chair	25.2	**25.3**	clock	**49.5**	49.0
couch	37.7	**39.4**	cow	51.3	**51.5**	cup	39.8	**40.1**
dining table	**25.3**	25.1	dog	57.5	**57.7**	donut	**42.6**	42.2
elephant	59.4	**60.9**	fire hydrant	**65.9**	65.0	fork	30.3	**30.8**
frisbee	62.9	**63.4**	giraffe	64.1	**65.5**	hair drier	0.5	**1.9**
handbag	11.8	**12.0**	horse	54.8	**54.9**	hot dog	27.2	**30.5**
keyboard	47.3	**48.4**	kite	39.8	**40.1**	knife	15.1	**15.5**
laptop	56.6	**57.2**	microwave	53.4	**54.3**	motorcycle	**40.8**	40.2
mouse	**62.8**	62.7	orange	**29.1**	28.2	oven	30.7	**31.5**
parking meter	44.9	**45.9**	person	53.6	**53.7**	pizza	49.4	**50.4**
potted plant	**24.6**	24.2	refrigerator	52.7	**53.5**	remote	26.9	**27.2**
sandwich	31.0	**31.8**	scissors	20.0	**21.3**	sheep	47.1	**48.0**
sink	34.2	**34.3**	skateboard	46.6	**46.7**	skis	**21.3**	20.3
snowboard	**33.9**	33.4	spoon	13.8	**14.0**	sports ball	47.4	**47.5**
stop sign	63.4	**64.7**	suitcase	33.9	**35.2**	surfboard	35.0	**35.5**
teddy bear	43.0	**44.7**	tennis racket	44.0	**44.6**	tie	31.8	**31.9**
toaster	39.0	**43.0**	toilet	**54.7**	54.5	toothbrush	20.0	**21.2**
traffic light	27.2	**27.7**	train	**58.4**	57.9	truck	32.0	**32.5**
tv	52.7	**54.4**	umbrella	35.8	**36.7**	vase	35.8	**36.1**
wine glass	**34.0**	33.9	zebra	64.0	**65.0**	**mAP**	38.6	**39.1**

Table 4.6: Instance segmentation AP results for calibration with image and instance counts on COCO.

Category	Baseline	PriorBox	Category	Baseline	PriorBox	Category	Baseline	PriorBox
airplane	49.0	**49.1**	apple	18.0	**18.3**	backpack	13.8	**14.1**
banana	**17.2**	17.1	baseball bat	20.1	**20.7**	baseball glove	36.9	**37.5**
bear	64.8	**65.9**	bed	**30.7**	30.6	bench	15.6	**15.8**
bicycle	16.3	16.3	bird	29.2	**29.8**	boat	22.0	**22.2**
book	**9.0**	8.9	bottle	36.2	**36.6**	bowl	38.2	**38.3**
broccoli	22.1	**22.6**	bus	61.7	61.7	cake	**34.2**	34.1
car	39.1	**39.5**	carrot	18.1	**18.5**	cat	65.3	**65.8**
cell phone	32.6	**33.1**	chair	16.4	16.4	clock	**50.0**	49.7
couch	32.8	**33.6**	cow	**44.0**	43.9	cup	40.4	**40.5**
dining table	**14.1**	14.0	dog	56.3	**56.7**	donut	**43.5**	43.1
elephant	55.6	**56.0**	fire hydrant	**63.1**	62.9	fork	**13.4**	13.1
frisbee	61.6	**61.8**	giraffe	**49.1**	49.0	hair drier	0	**0.3**
handbag	12.6	**12.7**	horse	39.2	**39.6**	hot dog	20.6	**22.7**
keyboard	47.3	**48.1**	kite	29.3	**29.7**	knife	9.0	9.0
laptop	**58.0**	57.9	microwave	**55.9**	55.5	motorcycle	**30.8**	30.7
mouse	**62.6**	62.4	orange	**29.0**	28.4	oven	28.8	**30.1**
parking meter	46.4	**47.2**	person	46.0	**46.1**	pizza	49.3	**50.1**
potted plant	**21.6**	21.4	refrigerator	53.8	**54.6**	remote	25.2	**25.3**
sandwich	32.4	**33.5**	scissors	15.1	**16.0**	sheep	41.1	**41.8**
sink	32.7	**32.9**	skateboard	27.3	27.3	skis	**2.0**	1.9
snowboard	20.3	20.3	spoon	**9.1**	8.8	sports ball	46.5	**46.7**
stop sign	64.3	**65.0**	suitcase	36.1	**36.8**	surfboard	28.9	**29.4**
teddy bear	42.5	**42.9**	tennis racket	52.3	52.3	tie	**30.3**	30.0
toaster	42.5	**46.5**	toilet	55.2	**55.8**	toothbrush	**13.6**	13.3
traffic light	26.0	**26.6**	train	57.6	**58.3**	truck	31.5	**32.3**
tv	55.1	**56.3**	umbrella	42.6	**42.7**	vase	34.7	**35.0**
wine glass	29.1	**29.9**	zebra	55.1	**55.6**	**mAP**	35.2	**35.6**

Table 4.7: Ground up training results. * denotes PriorBox model with image and instance count priors calibrating the logits.

	Detection				Segmentation			
Model	AP	AP_r	AP_c	AP_f	AP	AP_r	AP_c	AP_f
R50-FPN baseline	22.5	9.1	**21.1**	30.1	21.7	9.6	21.0	27.8
PriorBox Ground Up*	**23.6**	**12.0**	**21.1**	**30.5**	**22.8**	**13.4**	**21.7**	**28.1**

4.5.5 Training From the Ground Up

After analyzing PriorBox as a post-calibration technique, we train Mask R-CNN [9] models from the ground up with the original hyperparameters like learning rate, batch size and training schedule and the integrated priors. Having distributional prior during the training guides and speed-ups the learning process. PriorBox improves MMDetection [44] LVIS baseline[7] on all average precision metrics, mainly AP_r (Table 4.7).

4.5.6 Comparison with the Existing Methods

We first compare PriorBox with the existing post-calibration methods Table 4.8. Even though PriorBox is a simple calibration method, we also share a comparison with the state-of-the-art methods in Table 4.9.

4.6 Ablation Study

We conduct detailed experiments to analyze contributions from different PriorBox components. PriorBox has three fundamental ideas to investigate:

1. Fusing information from priors
2. Sorting class scores to exploit order statistics
3. Applying convolutional layers to competing classes

[7] https://github.com/open-mmlab/mmdetection/tree/master/configs/lvis#results-and-models-of-lvis-v1

Table 4.8: Comparison with the post-calibration methods. ResNet-50 backbone [34] on LVIS v1 with RFS. Results are retrieved from [33]

	Segmentation			
Model	AP	AP_r	AP_c	AP_f
RFS baseline [17]	22.6	12.3	21.3	28.6
HistBin [47]	21.8	11.3	20.3	28.1
BBQ (AIC) [48]	22.1	11.4	20.7	28.21
Beta calibration [49]	22.6	12.3	21.3	28.5
Isotonic reg. [50]	22.4	12.2	21.1	28.4
Platt scaling [51]	22.6	12.3	21.3	28.5
PriorBox img + ins	23.1	14.6	21.7	28.6
NorCal [33]	25.2	19.3	24.2	29.0

.

Table 4.9: Comparison with the state-of-the-art. ResNet-50 backbone [34] on LVIS v1 with RFS. ζ: implemented with Detectron2 [40]. ξ: implemented with MMDetection [44].

	Segmentation			
Model	AP	AP_r	AP_c	AP_f
RFS baseline [17] ζ	22.6	12.3	21.3	28.6
PriorBox img + ins ζ	23.1	14.6	21.7	28.6
EQL v2 [7] ξ	23.7	14.9	22.8	28.6
MosaicOS [42] ζ	24.5	18.2	23.0	28.8
RS Loss [24] ξ	25.2	16.8	24.3	29.9
NorCal [33] ζ	25.2	19.3	24.2	29.0
BAGS [21] ξ	26.2	18.0	26.9	28.7

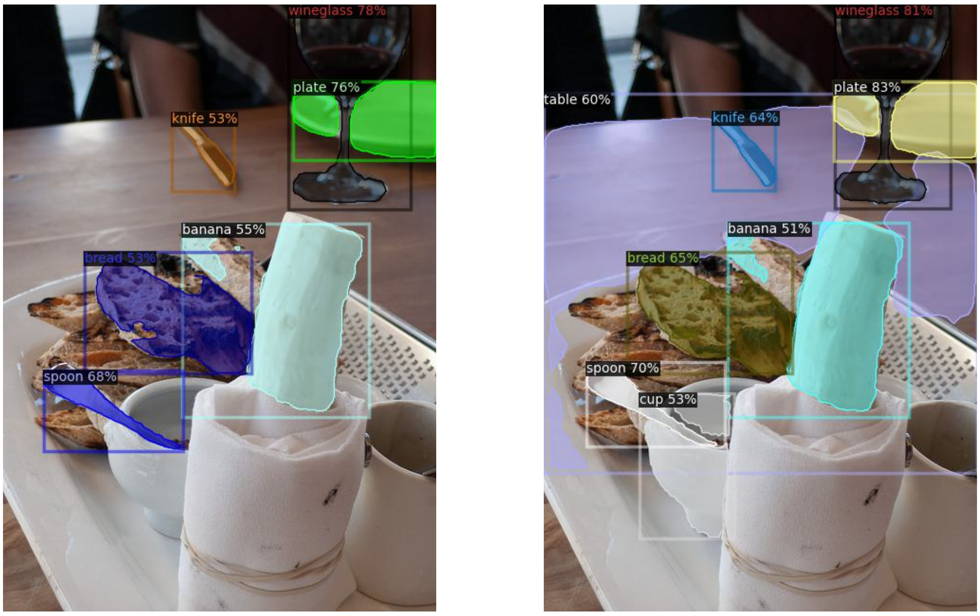


Figure 4.2: Comparison between RFS baseline [17] (left) and PriorBox (right). PriorBox predicts higher probabilities for true positives (wineglass, knife, plate, bread and spoon), detects more ground truths (table and cup) and gives less confidence to false-positives (banana) than the baseline. Image is picked from LVIS v1 validation split.

We train separate models with the same commonly used, re-producible **baseline** to have a decent comparison. We adopt Mask R-CNN [9] model with ResNet-50 [34] backbone as in Section 4.4, (1) in Table 4.10.

Since PriorBox calibrates pre-trained models, the number of epochs increases accordingly. **Longer training schedules** have the potential to improve model performance. Thus, we retrain the baseline model without modifying any layers and hyperparameters. As rare classes have fewer samples than the others, their results improved more in the retrained model, (2) in Table 4.10.

PriorBox rescores logits and probabilities with respect to their values and priors. On the other hand, **rescoring without priors** can be effective by adopting scores as learning status signals. We train the logit calibration network with (1x2) kernels and note that rescoring without priors performs similar to the re-trained baseline, (3) in Table 4.10. Surprisingly, **replacing (1x2) kernels with (1x1) ones**, (4) in Table 4.10, performs better. We note that class comparison works better with the guiding priors, especially on frequent classes, experiments (6) and (9) in Table 4.10.

Priors guide the training process to reduce the data imbalance effects. Adding **image and instance count priors (without sorting, with (1x2) kernels)** ignores the order statistics, but still enhances the detection and segmentation results on rare classes, (5) in Table 4.10. Adding sorting boosts all results, especially segmentation on rare classes, (9) in Table 4.10.

Having (1x2) kernels considers competing classes during the rescoring. If we use **(1x1) kernels with priors**, rescoring will handle each class separately. The (1x1) kernel model performs well on both tasks, especially on rare and common categories, (6) in Table 4.10.

As PriorBox calibrates outputs from pre-trained networks, the method is sensitive to the frozen layers. PriorBox can work with fully frozen networks, but the results differ drastically, it can't learn the calibration factors with the 3-layer convolutional network. We compare **freezing the base network fully and different ResNet stages**[8], (7) and (8) in Table 4.10. Updating ResNet weights during the calibration causes

[8] https://detectron2.readthedocs.io/en/latest/modules/modeling.html#detectron2.modeling.ResNet.freeze

Table 4.10: Ablation study results.

	Detection				Segmentation			
Model	AP	AP_r	AP_c	AP_f	AP	AP_r	AP_c	AP_f
(1) Baseline	23.3	12.3	21.3	30.2	22.6	12.3	21.3	28.6
(2) Baseline re-train	23.6	13.4	21.6	30.3	22.9	13.7	21.5	28.6
(3) No-priors with (1x2) kernels	23.6	13.2	21.6	30.3	22.9	13.7	21.5	28.6
(4) No-priors with (1x1) kernels	23.6	13.9	21.6	30.2	23.0	14.2	21.5	28.5
(5) Priors without sorting	23.6	13.9	21.6	30.3	23.0	14.1	21.5	28.6
(6) Priors with (1x1) kernels	23.7	14.0	21.7	30.2	23.1	14.3	21.6	28.5
(7) Freeze all layers	23.3	12.4	21.4	30.2	22.6	12.4	21.4	28.5
(8) Freeze 2 stages in ResNet	23.6	13.3	21.8	30.2	22.8	13.3	21.5	28.5
(9) Best	23.8	14.0	21.8	30.3	23.1	14.6	21.7	28.6

unlearning effect and results in worse predictions from the re-trained baseline.

The best model adopts image and instance count priors, sorted with respect to the logits, (1x2) kernels in a 3-layer convolutional network and 5 frozen stages in ResNet, (9) in Table 4.10.

CHAPTER 5

CONCLUSION

Today, training detectors in a low-sample regime is a central problem in expanding the use of deep learning applications in the real world. The natural distribution of images leads to data imbalance between the foreground classes and the overwhelming negative gradients suppress rare classes during the training. Subsequently, there is a significant gap between detection accuracies of head and tail categories. Re-sampling, re-weighting and grouping-based methods attempt to handle this imbalance, but they introduce additional hyperparameters and complex loss functions.

The proposed method, PriorBox, re-ranks the proposal scores and probabilities by extracting calibration factors from easy-to-collect priors (image count, instance count, size count and aspect ratio difference) and a simple 3-layer convolutional network. Comprehensive experiments confirm the method's ability to improve object detection and instance segmentation metrics without changing the data sampler and loss function (Figure 4.2). As PriorBox is easy to integrate into the existing architectures, it can be extended with new priors and adopted as a general calibration method. In the future, PriorBox can be utilized to predict bounding box offsets to improve instance localization.

REFERENCES

[1] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, “Imbalance problems in object detection: A review,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3388–3415, 2020.

[2] N. Chang, Z. Yu, Y.-X. Wang, A. Anandkumar, S. Fidler, and J. M. Alvarez, “Image-level or object-level? a tale of two resampling strategies for long-tailed detection,” in *International Conference on Machine Learning*, pp. 1463–1472, PMLR, 2021.

[3] J. Ren, C. Yu, X. Ma, H. Zhao, S. Yi, *et al.*, “Balanced meta-softmax for long-tailed visual recognition,” *Advances in neural information processing systems*, vol. 33, pp. 4175–4186, 2020.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “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.

[5] Y.-X. Wang, D. Ramanan, and M. Hebert, “Learning to model the tail,” *Advances in neural information processing systems*, vol. 30, 2017.

[6] J. Tan, C. Wang, B. Li, Q. Li, W. Ouyang, C. Yin, and J. Yan, “Equalization loss for long-tailed object recognition,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11662–11671, 2020.

[7] J. Tan, X. Lu, G. Zhang, C. Yin, and Q. Li, “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.

[8] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, “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.

[9] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[11] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[12] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[15] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[17] A. Gupta, P. Dollar, and R. Girshick, "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.

[18] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense

object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

[19] A. Krizhevsky, "Learning multiple layers of features from tiny images," tech. rep., Canadian Institute for Advanced Research, 2009.

[20] G. Van Horn, O. Mac Aodha, Y. Song, A. Shepard, H. Adam, P. Perona, and S. Belongie, "The inaturalist challenge 2017 dataset," *arXiv preprint arXiv:1707.06642*, vol. 1, no. 2, p. 4, 2017.

[21] Y. Li, T. Wang, B. Kang, S. Tang, C. Wang, J. Li, and J. Feng, "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.

[22] T. Wang, Y. Zhu, C. Zhao, W. Zeng, J. Wang, and M. Tang, "Adaptive class suppression loss for long-tail object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3103–3112, 2021.

[23] T. Wang, Y. Li, B. Kang, J. Li, J. Liew, S. Tang, S. Hoi, and J. Feng, "The devil is in classification: A simple framework for long-tail instance segmentation," in *European conference on computer vision*, pp. 728–744, Springer, 2020.

[24] K. Oksuz, B. C. Cam, E. Akbas, and S. Kalkan, "Rank & sort loss for object detection and instance segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3009–3018, 2021.

[25] D. Samuel, Y. Atzmon, and G. Chechik, "From generalized zero-shot learning to long-tail with class descriptors," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 286–295, 2021.

[26] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 951–958, IEEE, 2009.

[27] Y. Xian, B. Schiele, and Z. Akata, "Zero-shot learning-the good, the bad and the ugly," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4582–4591, 2017.

[28] Y. Atzmon and G. Chechik, "Probabilistic and-or attribute grouping for zero-shot learning," *arXiv preprint arXiv:1806.02664*, 2018.

[29] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," tech. rep., California Institute of Technology, 2011.

[30] G. Patterson and J. Hays, "Sun attribute database: Discovering, annotating, and recognizing scene attributes," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2751–2758, IEEE, 2012.

[31] A. Dave, P. Dollár, D. Ramanan, A. Kirillov, and R. Girshick, "Evaluating large-vocabulary object detectors: The devil is in the details," *arXiv preprint arXiv:2102.01066*, 2021.

[32] A. Jadbabaie, A. Makur, and D. Shah, "Federated optimization of smooth loss functions," *arXiv preprint arXiv:2201.01954*, 2022.

[33] T.-Y. Pan, C. Zhang, Y. Li, H. Hu, D. Xuan, S. Changpinyo, B. Gong, and W.-L. Chao, "On model calibration for long-tailed object detection and instance segmentation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 2529–2542, 2021.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[35] R. G. Cinbis and S. Sclaroff, "Contextual object detection using set-based classification," in *European Conference on Computer Vision*, pp. 43–57, Springer, 2012.

[36] L. V. Pato, R. Negrinho, and P. M. Aguiar, "Seeing without looking: Contextual rescoring of object detections for ap maximization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14610–14618, 2020.

[37] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014.

[38] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[39] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[40] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2." `https://github.com/facebookresearch/detectron2`, 2019.

[41] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

[42] C. Zhang, T.-Y. Pan, Y. Li, H. Hu, D. Xuan, S. Changpinyo, B. Gong, and W.-L. Chao, "Mosaicos: a simple and effective use of object-centric images for long-tailed object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 417–427, 2021.

[43] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "{TensorFlow}: a system for {Large-Scale} machine learning," in *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pp. 265–283, 2016.

[44] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, *et al.*, "Mmdetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.

[45] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra, "Detecting twenty-thousand classes using image-level supervision," *arXiv preprint arXiv:2201.02605*, 2022.

[46] C. Zhang, T.-Y. Pan, T. Chen, J. Zhong, W. Fu, and W.-L. Chao, "Learning with free object segments for long-tailed instance segmentation," *arXiv preprint arXiv:2202.11124*, 2022.

[47] B. Zadrozny and C. Elkan, "Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers," in *Icml*, vol. 1, pp. 609–616, Citeseer, 2001.

[48] M. P. Naeini, G. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using bayesian binning," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[49] M. Kull, T. Silva Filho, and P. Flach, "Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers," in *Artificial Intelligence and Statistics*, pp. 623–631, PMLR, 2017.

[50] B. Zadrozny and C. Elkan, "Transforming classifier scores into accurate multiclass probability estimates," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 694–699, 2002.

[51] J. Platt *et al.*, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.