USING A RANKING-BASED LOSS FOR LONG-TAILED VISUAL RECOGNITION

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

BY

BARAN GÜLMEZ

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

APRIL 2024

Approval of the thesis:

USING A RANKING-BASED LOSS FOR LONG-TAILED VISUAL RECOGNITION

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

Prof. Dr. Naci Emre Altun
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Mehmet Halit S. Oğuztüzün
Head of Department, **Computer Engineering** _____

Prof. Dr. Sinan Kalkan
Supervisor, **Computer Engineering, METU** _____

Assoc. Prof. Dr. Emre Akbaş
Co-supervisor, **Computer Engineering, METU** _____

Examining Committee Members:

Assoc. Prof. Dr. Ramazan Gökberk Cinbiş
Computer Engineering, METU _____

Prof. Dr. Sinan Kalkan
Computer Engineering, METU _____

Prof. Dr. Ahmet Burak Can
Computer Engineering, Hacettepe _____

Date: 16.04.2024

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: Baran Gülmez

Signature :

ABSTRACT

USING A RANKING-BASED LOSS FOR LONG-TAILED VISUAL RECOGNITION

Gülmez, Baran
M.S., Department of Computer Engineering
Supervisor: Prof. Dr. Sinan Kalkan
Co-Supervisor: Assoc. Prof. Dr. Emre Akbaş

APRIL 2024, 43 pages

Long-tailed visual recognition, where certain classes contain far fewer samples than others, poses a critical challenge in learning-based computer vision applications. As real-world visual recognition datasets generally exhibit long-tailed distributions, addressing the challenge of learning in such long-tailed datasets is essential for many applications. In this thesis, for long-tailed visual recognition, we explore and adapt the Average Precision (AP) Loss, which was originally proposed by Chen et al. for the task of object detection. We found that the standard AP Loss performs similarly to traditional loss functions like Cross Entropy Loss on dealing with uneven class distributions. By introducing two specific modifications to AP Loss, we significantly improved the model's accuracy in identifying rare classes and its overall performance across all classes. We conducted thorough experiments to compare these improved AP Loss versions with other top-performing loss functions in the literature. Our findings showed that our modified AP Loss versions provide on par with or better performance than state-of-the-art loss functions.

Keywords: Long-tailed Visual Recognition, Ranking Based Loss Functions

ÖZ

UZUN KUYRUKLU GÖRSEL TANIMA İÇİN SIRALAMA TABANLI KAYIP FONKSİYONU KULLANIMI

Gülmez, Baran
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Sinan Kalkan
Ortak Tez Yöneticisi: Doç. Dr. Emre Akbaş

2024, 43 sayfa

Uzun kuyruklu görsel tanıma, yani bazı sınıfların diğerlerinden çok daha az örneğe sahip olması durumu, öğrenme tabanlı bilgisayarlı görü uygulamalarında kritik bir zorluk oluşturmaktadır. Gerçek dünya görsel tanıma veri setleri genellikle uzun kuyruklu dağılımlar sergilediğinden, bu tür veri setlerinde öğrenme zorluğunun üstesinden gelmek birçok uygulama için çok önemlidir. Bu tezde, uzun kuyruklu görsel tanıma için, Chen ve arkadaşları tarafından orijinal olarak nesne tespiti görevi için önerilen Ortalama Kesinlik (AP) Kaybını inceliyor ve uyarlıyoruz. Standart AP Kaybının, dengesiz sınıf dağılımlarıyla başa çıkmada Çapraz Entropi Kaybı gibi geleneksel kayıp fonksiyonlarına benzer şekilde performans gösterdiğini bulduk. AP Kaybına iki özel değişiklik getirerek, modelin nadir sınıfları tanıma doğruluğunu ve tüm sınıflar arasındaki genel performansını önemli ölçüde iyileştirdik. Bu iyileştirilmiş AP Kaybı versiyonlarını literatürdeki diğer en iyi performans gösteren kayıp fonksiyonlarıyla karşılaştırmak için kapsamlı deneyler yürüttük. Bulgularımız, geliştirdiğimiz AP Kaybı versiyonlarının, diğer en iyi performans gösteren kayıp fonksiyonlarıyla

eşdeğer veya onlardan daha iyi performans sunduğunu gösterdi.

Anahtar Kelimeler: Uzun Kuyruklu Görüntü Tanıma, Sıralama Tabanlı Kayıp Fonksiyonları

This thesis is dedicated to my family.

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my advisors, Prof. Dr. Sinan Kalkan and Assoc. Prof. Dr. Emre Akbaş, for their guidance throughout my MSc studies, as well as for their patience, motivation, and invaluable insights. I also owe a debt of gratitude to our research team, most notably to Dr. Kemal Öksüz, for their invaluable input.

It was my brother and my friends who made the hardships look manageable.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xi

LIST OF TABLES xiii

LIST OF FIGURES xv

LIST OF ABBREVIATIONS xvi

CHAPTERS

1 INTRODUCTION 1

1.1 Motivation 1

1.2 Problem Definition and Scope of the Thesis 2

1.3 Contributions 4

1.4 The Outline of the Thesis 5

2 BACKGROUND AND RELATED WORK 7

2.1 Long-Tailed Visual Recognition 7

2.1.1 Data Processing 7

2.1.2 Cost Sensitive Weighting 9

2.1.2.1 Class-level Re-weighting 9

2.1.2.2 Instance-level Re-weighting 10
2.1.3 Decoupling Methods 10
2.1.4 Other Methods 11
2.2 AP Loss Overview and Its Variants 12
2.2.1 AP Loss Variants 14
2.3 Datasets 16
3 METHODS 19
3.1 Revisiting Ranking-based Losses 19
3.2 Proposed Extension 1: Class Balancing for Delta Parameter 23
3.3 Proposed Extension 2: Class Balancing for Primary Term 24
4 EXPERIMENTS 27
4.1 Training and Implementation Details 27
4.2 Experiments on Proposed Extension 1: Modifying Delta Parameter . 27
4.3 Experiments on Balancing Strategy: Testing Progressive Balancing . 29
4.4 Comparison with Other Methods 32
5 CONCLUSION AND FUTURE WORK 37
5.1 Conclusion 37
5.2 Future Work 38
REFERENCES 39

LIST OF TABLES

TABLES

Table 4.1 CIFAR-10 Imbalance Ratio 50 accuracies for delta multiplication and division. Best results are shown in bold and the second best results are shown with *. 28

Table 4.2 CIFAR-10 Imbalance Ratio 100 accuracies for delta multiplication and division. Best results are shown in bold and the second best results are shown with *. 28

Table 4.3 CIFAR-100 Imbalance Ratio 50 accuracies for delta multiplication and division. Best results are shown in bold and the second best results are shown with *. 29

Table 4.4 CIFAR-100 Imbalance Ratio 100 accuracies for delta multiplication and division. Best results are shown in bold and the second best results are shown with *. 29

Table 4.5 CIFAR-10 Imbalance Ratio 50 accuracies for comparing progressive balancing. Best results are shown in bold and the second best results are shown with *. 30

Table 4.6 CIFAR-10 Imbalance Ratio 100 accuracies for comparing progressive balancing. Best results are shown in bold and the second best results are shown with *. 30

Table 4.7 CIFAR-100 Imbalance Ratio 50 accuracies for comparing progressive balancing. Best results are shown in bold and the second best results are shown with *. 31

Table 4.8 CIFAR-100 Imbalance Ratio 100 accuracies for comparing progressive balancing. Best results are shown in bold and the second best results are shown with *. 31

Table 4.9 CIFAR-10 Imbalance Ratio 50 accuracies for comparison with other methods. The results with '†' are retrieved from Baltaci et al. [1]. Best results are show with bold and the second best results are shown with '*'. . 33

Table 4.10 CIFAR-10 Imbalance Ratio 100 accuracies for comparison with other methods. The results with '†' are retrieved from Baltaci et al. [1]. Best results are show with bold and the second best results are shown with '*'. 34

Table 4.11 CIFAR-100 Imbalance Ratio 50 accuracies for comparison with other methods. The results with '†' are retrieved from Baltaci et al. [1]. Best results are show with bold and the second best results are shown with '*'. 35

Table 4.12 CIFAR-100 Imbalance Ratio 100 accuracies for comparison with other methods. The results with '†' are retrieved from Baltaci et al. [1]. Best results are show with bold and the second best results are shown with '*'. 35

LIST OF FIGURES

FIGURES

Figure 1.1 Long-tailed class distribution, illustrating the disparity between head and tail classes. Figure source is [2]. 2

Figure 2.1 Categorization of Long-Tailed Visual Recognition methods, adapted from Yang et al. [3]. 8

Figure 2.2 Training and Test Set Class Distribution. The figure's left side illustrates the training set's long-tailed distribution, characterized by a decreasing number of samples per class, resulting in class imbalance. The right side illustrates the balanced distribution of the test set, with each class having an equal number of samples. Figure source is [3]. . . 16

Figure 3.1 Ranking based losses computation (rightward arrows) formulation by Oksuz et al. [4]. The leftward arrows show the optimization part. Figure source is [4]. 20

Figure 3.2 Different values of piecewise linear function (Equation 3.5) for different δ. Figure source is [5]. 24

LIST OF ABBREVIATIONS

AP	Average precision
CIFAR	Canadian Institute For Advanced Research
IR	Imbalance Ratio
CB	Class Based
LDAM	Label-Distribution-Aware Margin Loss
LT	Long-Tailed
DRO-LT	Distributional Robustness Loss

CHAPTER 1

INTRODUCTION

1.1 Motivation

Visual recognition is one of the landmark problems in computer vision, with increasingly widespread applications in different domains in recent years [6, 7, 8]. This surge in interest is partly due to technological advancements and the unprecedented availability of data. Visual recognition is now integral in various industries, finding applications in retail for analyzing customer behavior [9], in remote sensing for Global Information Systems [10], in the medical field for disease detection [11], and in the automotive industry for autonomous driving [12]. As the demand in these areas has grown, so has the volume of the data used, presenting unique challenges in data handling and analysis.

This vast, real-world data is inherently imbalanced in many aspects, including class, scale, spatial, or objective imbalances [13]. We specifically focus on imbalances in the number of samples across different object types, known as class cardinality. A significant variance in the number of samples per class, often referred to as class imbalance, or long-tailedness, is a recognized issue in the literature [14], but remains unresolved.

Although imbalance ratios in various domains vary, they commonly exhibit a pattern where data is skewed exponentially rather than linearly (Figure 1.1). This skewness manifests in the distribution of class examples, leading to a clear division into 'head classes' and 'tail classes.' Head classes are those with a significantly larger number of examples, typically representing common objects frequently encountered in specific domains. Their over-representation often results in models that are biased towards

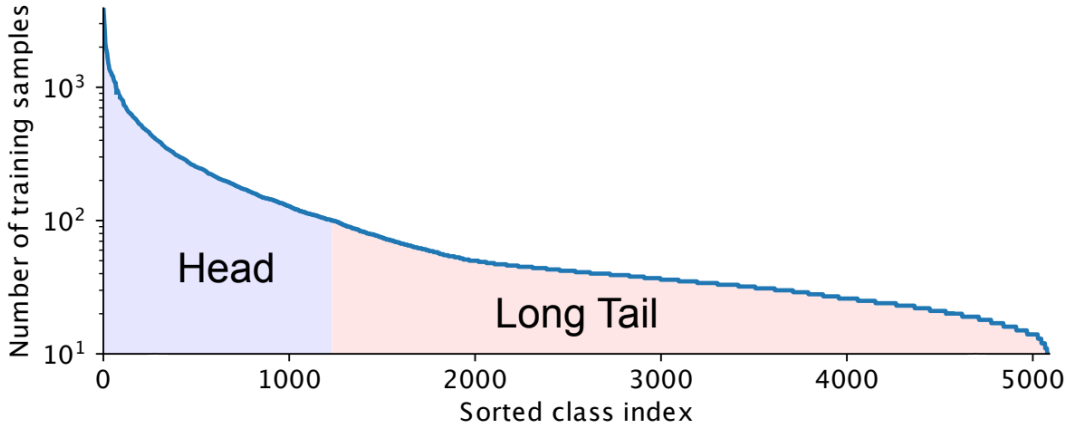


Figure 1.1: Long-tailed class distribution, illustrating the disparity between head and tail classes. Figure source is [2].

these classes. Conversely, tail classes are characterized by far fewer instances and include rare or less commonly seen objects. This disparity in class representation poses a significant challenge in developing unbiased and effective visual recognition systems. Figure 1.1 illustrates this long-tailed distribution, visually depicting the stark contrast between head and tail classes.

1.2 Problem Definition and Scope of the Thesis

Visual recognition is nowadays addressed by training a model to predict labels in a dataset with imbalanced class distributions. Consider a sample (x, y) from a dataset D, which comprises C classes and N samples. In the context of visual recognition, x represents the input image, and y, the correct label from the set $\{1, \ldots, C\}$. The quantity of samples in a class c, where $c \in \{1, \ldots, C\}$, is indicated by n_c. Let us denote the count of samples in the most populous class with $n_{\text{max}} = \max\{n_1, n_2, \ldots, n_C\}$ and in the least populous class as $n_{\text{min}} = \min\{n_1, n_2, \ldots, n_C\}$. Generally, the disparity among class cardinalities is represented by the so-called imbalance ratio, which is defined as $IR = n_{\text{max}}/n_{\text{min}}$ [2].

In visual recognition, a model is trained to map the input image x to a probability distribution over the C classes. This mapping is typically achieved through a neural network, which outputs a vector of logits, $z \in \mathbb{R}^C$, representing the unnormalized log

probabilities for each class. The logits are converted into a probability distribution through the softmax function:

$$P(y = c|x) = \frac{e^{z_c}}{\sum_{j=1}^{C} e^{z_j}}, \tag{1.1}$$

where $P(y = c|x)$ denotes the likelihood that the image x is categorized as class c.

The training aims to adjust the model's parameters so that its predicted probability distribution is closely aligned with the actual label distribution in the dataset. This objective is accomplished through the minimization of a loss function. In visual recognition, Cross Entropy Loss is frequently employed. This loss quantifies the disparity between two probability distributions, the predicted and the true distributions. For an individual training sample (x, y), Cross Entropy Loss is expressed as:

$$\mathcal{L}_{CE}(x, y) = -\log P(y|x) = -\log\left(\frac{e^{z_y}}{\sum_{j=1}^{C} e^{z_j}}\right). \tag{1.2}$$

The loss function is designed to prompt the model to assign a higher likelihood to the correct class while reducing probabilities for the other classes.

Throughout the training phase, the model parameters undergo iterative updates to reduce Cross Entropy loss, which is averaged across all training instances. This procedure generally employs gradient-based techniques such as Stochastic Gradient Descent (SGD) or its variations.

Average Precision (AP) Loss, introduced by Chen et al. [5], is a ranking-based loss function designed for object detection tasks. It directly optimizes the AP metric, which is the primary evaluation metric for object detection. AP Loss assigns a ranking error to each positive example based on its rank among all examples and optimizes the model to minimize these errors. In this thesis, we propose two extensions to the AP Loss to address the challenges posed by long-tailed class distributions in object detection, motivated by their proven robustness to positive-negative imbalance [4].

The first extension, 'Class Balancing for Delta Parameter' (Section 3.2), incorporates class-aware adjustments to the delta parameter within the AP Loss. The delta parameter controls the shape of the piecewise linear function used to establish the ranking relation between examples. By assigning higher delta values to under-represented

classes and lower delta values to over-represented classes, we aim to give more importance to the tail classes during training.

The second extension, 'Class Balancing for Primary Term' (Section 3.3), directly modifies the primary term of the AP Loss. The primary term is responsible for assigning ranking errors to positive examples based on their ranks. We multiply the primary term by class-specific weights, which are computed using the effective number of samples for each class. This approach ensures that the loss function pays more attention to the tail classes, despite their smaller number of instances.

Both proposed extensions utilize the class weighting strategy introduced by Cui et al. [2], which assigns higher weights to under-represented classes and lower weights to over-represented classes. The weights are determined based on the sorted class index and a hyperparameter β.

Through these class-balancing extensions, we aim to improve the AP Loss's performance on long-tailed object detection datasets, where a few classes dominate the data while many classes have limited representation. By focusing more on the tail classes during training, we expect the model to learn a more balanced representation and achieve better overall performance.

1.3 Contributions

Our contributions can be outlined as follows:

1. Motivated by the study of Oksuz et al. [4], who proved that ranking-based loss functions naturally provide balance between the gradients of positive and negative samples even under severe imbalance ratios for the task of object detection, we explore the use of such a ranking-based loss function, namely AP Loss [4], for long-tailed visual recognition.

2. We explore two novel extensions over AP Loss specifically for a long-tailed setting: (i) Adjusting AP Loss's only hyperparameter differently for each class according to class cardinality. (ii) Loss reweighting with class cardinality.

3. We show that the introduced extensions provide significant boosts over Cross Entropy Loss and the vanilla AP Loss and performs on par with or better than state-of-the-art methods.

1.4 The Outline of the Thesis

The structure of the thesis is arranged as follows: Chapter 2 offers a comprehensive review of the background and relevant literature pertinent to our study. In Chapter 3, we provide an explanation of the dataset used in our experiments and detail our methods for proceeding. Chapter 4 is dedicated to the execution of our experiments. Next, Chapter 5 contains the concluding remarks.

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 Long-Tailed Visual Recognition

Long-tailed visual recognition deals with the challenge of accurately classifying images from datasets that exhibit a highly imbalanced distribution of samples across different classes. In such scenarios, a few classes (the 'head') have many examples, while a significant number of classes (the 'tail') are underrepresented. The primary challenge in long-tailed visual recognition is the model's bias towards head classes due to their overrepresentation in the training data. This leads to performance degradation for tail classes, as the model fails to learn their features effectively. Furthermore, the scarcity of data in tail classes makes it difficult to use traditional data augmentation techniques effectively.

Current methodologies in long-tailed visual recognition vary, with strategies focusing on representation learning and model adaptation. We use Yang et al.'s taxonomy in this section as decipted in Figure 2.1. Some approaches involve re-sampling the dataset, re-weighting the loss function, or employing transfer learning to augment the feature space of underrepresented classes. However, these methods often struggle with overfitting for tail classes and may not fully address the complex nature of long-tailed distributions.

2.1.1 Data Processing

Preprocessing the data to obtain more balanced class representations is a straightforward approach to addressing the challenges posed by long-tailed distributions. Yang

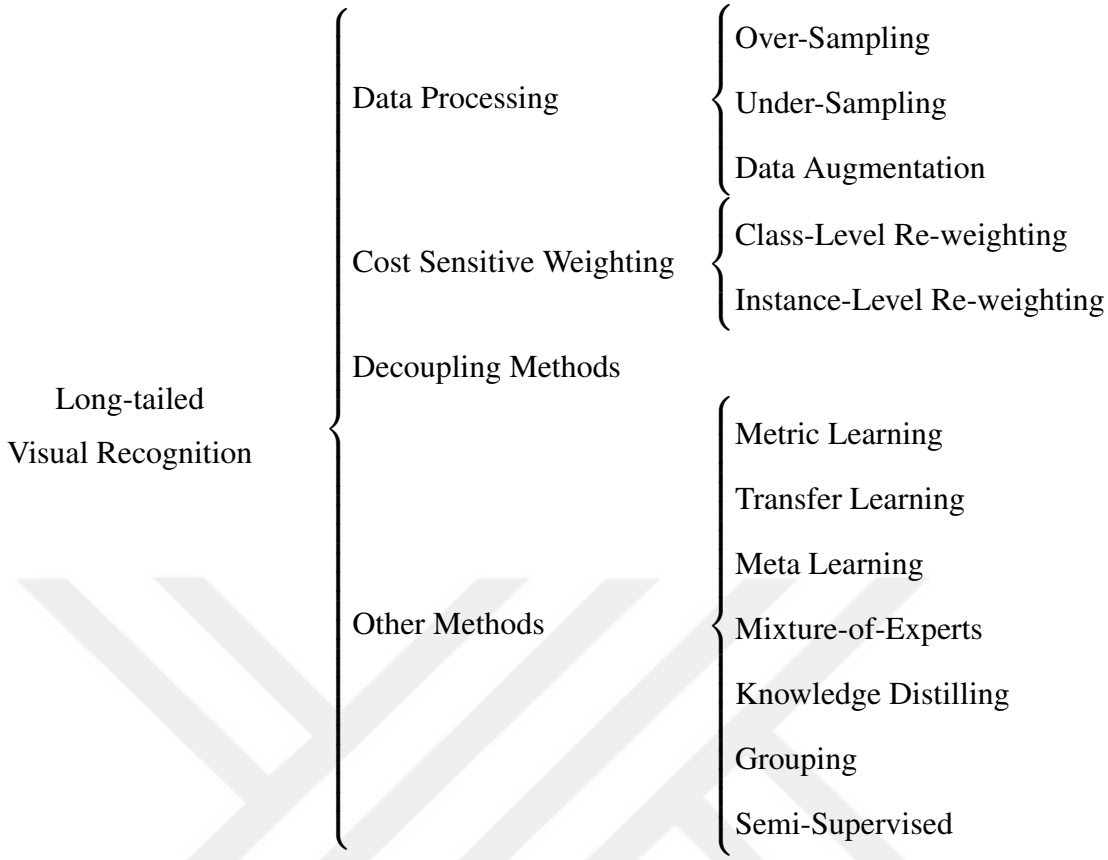


Figure 2.1: Categorization of Long-Tailed Visual Recognition methods, adapted from Yang et al. [3].

et al. [3] group data processing techniques for long-tailed recognition into three primary categories: over-sampling, under-sampling, and data augmentation.

Over-sampling techniques aim to increase the representation of tail classes by sampling more instances from them [15, 16]. This can be accomplished by duplicating tail class images at the image level or by repeating only the tail class annotations at the instance level. However, over-sampling may lead to overfitting models to the tail classes.

Under-sampling methods, on the other hand, remove instances from the head classes to improve class balance [17, 18, 19]. Although simple, under-sampling may discard valuable data and cause underfitting of head classes.

Data augmentation helps balance the dataset by synthetically generating additional training samples for the underrepresented tail classes. Traditional methods interpolate new samples between existing ones in either data or feature space. Recent approaches capitalize on advancements in neural generative models to synthesize more diverse and realistic samples. However, data augmentation cannot entirely bridge the gap between real and generated data distributions.

While data processing methods provide a straightforward approach to long-tailed recognition, they have limitations. Over-sampling and under-sampling inherently trade off performance between head and tail classes. Data augmentation creates additional training examples by applying transformations to existing data, helping to mitigate these trade-offs [20, 21]. Despite these limitations, data processing remains a widely-used tool, particularly when combined with other algorithmic approaches.

2.1.2 Cost Sensitive Weighting

Cost sensitive learning [22, 23] assigns different misclassification costs to classes or samples and is naturally applicable to long-tailed recognition due to its strong connection to imbalanced learning.

2.1.2.1 Class-level Re-weighting

Straightforward class-level re-weighting based on inverse class frequencies often performs poorly. Smoothed weights inversely proportional to the square root of label frequency are sometimes used. However, setting weights directly based on class sizes is often ineffective [**?**, 24]. Instead, methods implicitly distinguish head and tail classes based on factors like effective number of samples, gradient statistics, and sample difficulty [25]. Various approaches weight the loss term for each class differently, ignore or rebalance certain gradients, or dynamically adjust penalties.

2.1.2.2 Instance-level Re-weighting

Instance-level re-weighting can boost tail class performance by focusing on hard examples, which are prevalent in tail classes due to data imbalance [26, 24]. Methods include constructing high-loss mini-batches, modulating the loss to focus on hard examples, weighting gradients based on their distribution, and introducing instance-specific adjustment terms.

While cost sensitive re-weighting is an important strategy, it has limitations. Simple weighting schemes are often ineffective, finding suitable weights requires effort, instance-level weighting may still focus on head classes, optimization can be difficult for large-scale data, and hyper-parameters are sensitive.

2.1.3 Decoupling Methods

These methods separate the training process into two parts: learning representations and learning classifiers [27, 28, 29]. This separation is useful because trying to balance the data too early can negatively impact the quality of the data representations learned by the model.

In practice, the initial training uses uniform sampling to capture a broad set of features across all classes without prioritizing any specific class. This foundational phase is crucial for developing robust features that are not skewed towards any particular group of classes. Once this base of features is established, the training focus shifts to the classifier. At this stage, efforts are concentrated on achieving a balance between the overrepresented (head) and underrepresented (tail) classes, enhancing the model's ability to accurately classify less frequent classes without losing its effectiveness on more common ones.

Despite their effectiveness, decoupling methods move away from the streamlined end-to-end training typical in deep learning. They require careful management of the transition between the two training stages and can complicate the training process. Also, the rebalancing techniques used in the second stage, like re-sampling or re-weighting, still need to be handled with care to avoid introducing new biases or

training issues.

Despite these challenges, decoupling proves to be a potent strategy for tackling imbalanced datasets, offering a way to integrate it with other data processing and cost-sensitive weighting techniques to optimize learning outcomes. However, it's important to manage these methods thoughtfully to fully benefit from their potential without complicating the training process.

2.1.4 Other Methods

In addition to the aforementioned approaches, various other machine learning techniques have been researched to address the long-tailed recognition problem. These methods include metric learning, transfer learning, meta learning, knowledge distillation, mixture-of-experts, grouping, and semi-supervised learning.

Metric learning approach focuses on distinguishing between classes more clearly by enhancing the distances between differing class features while reducing the variation within the same class [30]. This method helps improve the clarity of class boundaries in long-tailed distributions, particularly benefiting the representation of minority classes.

In transfer learning, leveraging the abundance of data in frequent classes, transfer learning methods utilize the rich information available in these 'head' classes to boost the learning of under-represented 'tail' classes [31]. This strategy helps in transferring valuable knowledge from classes with ample data to those with scarce data.

Meta learning, also known as 'learning to learn',, adapts the model to new and varied tasks using knowledge gained from different tasks [32]. In the context of long-tailed distributions, it helps the model adjust its learning strategy to better handle classes with fewer examples by using meta-knowledge or meta-data.

Knowledge distillation technique involves a 'teacher-student' model setup where a more complex teacher model guides a simpler student model. The aim is to balance the learning between over-represented and under-represented classes, making the student model's predictions more equitable across different classes[33, 34].

Grouping methods divide the classes into subsets based on their instance numbers or semantic relationships, and train the model separately for each group[35].

Semi-supervised learning generates pseudo-labels for unlabeled data to expand the tail classes. This approach compensates for the insufficient representation of tail classes, but requires additional training data[36].

Mixture of Experts (MoE), involves training multiple expert models, each specialized for different segments of the dataset. This method manages to effectively address the disparities between the head and tail classes by allocating specific experts to handle different data complexities and distributions[37, 38].

While these methods have shown promise in addressing the long-tailed recognition problem, they also have their limitations. Metric learning and transfer learning may still struggle with extremely imbalanced data. Meta learning and knowledge distillation require careful design of meta-models or teacher-student architectures. MoE methods can be computationally expensive due to the need for training multiple expert models. Grouping strategies may limit knowledge sharing between groups, and semi-supervised learning relies on the availability of additional unlabeled data. Despite these challenges, the diversity of approaches demonstrates the active research efforts in tackling the long-tailed recognition problem, and future work may benefit from combining the strengths of different methods.

2.2 AP Loss Overview and Its Variants

The Average Precision loss (AP Loss) [5] was first formulated for single-stage object detectors to tackle the severe imbalance between foreground and background classes. In traditional object detection in computer vision, there are two main tasks: locating objects and identifying them. These tasks are typically handled through a multi-task framework, using separate loss functions for classification and localization. Single-stage detectors identify object classes immediately from pre-set candidate boxes (anchors), in contrast to two-stage detectors that first create object box proposals through category-neutral techniques, and then proceed to classification and localization.

In single-stage detectors, the classification component frequently faces a significant imbalance between foreground (object) and background (non-object) classes. This imbalance stems from the large number of anchors, which leads to a bias towards background classification in the learning process.

While AP-loss was initially designed for object detection tasks, its application has been extended to visual recognition. In visual recognition, the focus shifts from detecting objects within images to categorizing images into various classes. This development is important because it shows that AP-loss can be used in areas beyond its initial use, effectively tackling class imbalance problems in various visual recognition tasks.

AP-loss introduces a change in the framework, swapping the classification task with a ranking task. Using a ranking-focused loss, this approach better handles class imbalance by efficiently modeling how different samples relate to and are distributed among each other. The AP metric evaluates detection performance by looking at precision and recall across various thresholds, which makes it more resistant to the high number of true negatives that are common in standard classification metrics.

As a brief introduction for subsequent chapters, the following provides a concise overview of the AP Loss. The set of anchor boxes used in the object detection model is denoted by B. Each b_i within this set represents the i-th anchor box, serving as a candidate region for potential object locations. Each anchor box b_i is assigned a label t_i, categorizing it as positive (containing an object, denoted by $t_i = 1$), negative (part of the background, indicated by $t_i = 0$), or ignored (excluded from the loss calculation). Thus, sets of positive and negative samples are defined as $\mathcal{P} = \{i|t_i = 1\}$ and $\mathcal{N} = \{i|t_i = 0\}$, respectively. The score s_i, computed for each anchor box by the detection model, evaluates the probability of the box containing an object. The weights $\boldsymbol{\theta}$ for the network indicate the neural network parameters used to calculate the scores s_i for the anchor boxes.

The definition of AP (Average Precision) Loss is straightforward:

$$\mathcal{L}_{AP} = 1 - \text{AP} = 1 - \frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} \frac{\text{rank}^+(i)}{\text{rank}(i)}. \tag{2.1}$$

For calculating ranks, a comparison of the samples is performed using the Heaviside

step function:

$$H(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}. \tag{2.2}$$

The comparison metric is the score difference between anchor boxes:

$$\forall i, j, \quad x_{ij} = -(s(b_i; \boldsymbol{\theta}) - s(b_j; \boldsymbol{\theta})) = -(s_i - s_j). \tag{2.3}$$

Using this comparison function and the score difference, Chen et al. [5] define the primary term of the loss function as:

$$L_{ij}(\boldsymbol{x}) = \frac{H\left(x_{ij}\right)}{1 + \sum_{k \in \mathcal{P} \cup \mathcal{N}, k \neq i} H\left(x_{ik}\right)} = L_{ij}. \tag{2.4}$$

Finally, the loss can be defined using the primary term:

$$\mathcal{L}_{AP} = \frac{1}{|\mathcal{P}|} \sum_{iin\mathcal{P}} \sum_{jin\mathcal{N}} L_{ij} = \frac{1}{|\mathcal{P}|} \sum_{i,j} L_{ij} \cdot y_{ij}. \tag{2.5}$$

AP-loss presents challenges due to its non-differentiable and complex nature, making it unsuitable for direct optimization with standard gradient descent methods. However, Chen et al. [5] developed a unique optimization algorithm that combines the error-driven update method from perceptron learning with the back-propagation technique used in deep learning. This solution effectively optimizes the AP-loss, proving to be more effective in addressing class imbalance issues in object detection.

To sum up, AP-loss, originally designed for object detection, also offers a flexible and potent approach to address class imbalance in visual recognition. By replacing the classification task with a ranking task and applying a new optimization algorithm, AP-loss not only addresses the imbalance issue more effectively. This method paves the way for new research and practical applications within the realm of computer vision and object detection.

2.2.1 AP Loss Variants

Since its introduction, Average Precision loss (AP-loss) has significantly impacted object detection and visual recognition, primarily through its innovative approach in

tackling class imbalance by focusing on ranking over classification. This methodological shift has not only spurred extensive research but also led to the development of various AP-loss variants, each designed to address specific challenges in their respective areas. These variants demonstrate the evolution and diversification of the AP-loss concept, and in this section, we will elaborate on these notable variations, examining how different versions enhance and build upon the original framework to address distinct challenges in visual recognition.

In [39], the authors introduce significant advancements in dense object detection with Adaptive Pairwise Error (APE) and Adaptive Ranking Pair Selection (ARPS), enhancing Average Precision (AP) loss. These methods notably improve ranking pair selection between positive and negative samples. APE loss employs normalized ranking scores, combined with a clustering algorithm and localization scores, for enhanced accuracy in pairing and error determination, thus advancing the AP loss method. ARPS complements this by dynamically selecting negative samples based on their localization qualities, addressing localization accuracy disparities among positives. This method offers a comprehensive solution to pair selection challenges in dense object detection. APE and ARPS together effectively tackle the common problem of imbalance in this area, considerably improving the AP loss framework to meet the complex needs of dense object detection tasks.

In their study, [40] addresses the common issue of misclassification and missed detections of rare categories in object detectors dealing with extensive vocabularies and skewed label distributions. Traditional detectors often erroneously classify infrequent objects as common ones and tend to exclude lower-ranked detections, adversely affecting the detection of rare categories. To combat this, the paper introduces a dual-task learning framework named Reconciling Object-level and Global-level (ROG), designed to concurrently train models for both individual object classification and the global ranking of confidence scores. A key innovation within this framework is the Generalized Average Precision (GAP) loss. This loss function is strategically developed to correct ranking discrepancies among various objects across categories, ensuring an equitable distribution of gradients and enhancing the efficiency of the ranking process.

2.3 Datasets

The CIFAR-10 and CIFAR-100 datasets stand as widely used benchmarks in visual recognition. CIFAR-10 includes 60,000 color images, each 32x32 pixels, distributed across 10 classes, with each class comprising 6,000 images. In contrast, CIFAR-100, while similar in structure, features 100 classes, each with 600 images. These datasets are commonly used for assessing image classification algorithms, attributed to their varied and intricate image compositions.

To investigate the impact of class imbalance on visual recognition, we have created long-tailed versions of the CIFAR-10 and CIFAR-100 datasets using methodologies outlined in [41] and [2]. This modification involves altering the distribution of training images across different classes to mimic a long-tailed distribution, a common real-world scenario. Specifically, we adopt the exponential imbalance approach from [2] to create these long-tailed versions. The modification process entailed reducing the number of training samples per class according to predefined imbalance ratios, thereby creating datasets with varying degrees of class imbalance (refer to Figure 2.2 for a visual representation of training and test set distributions).

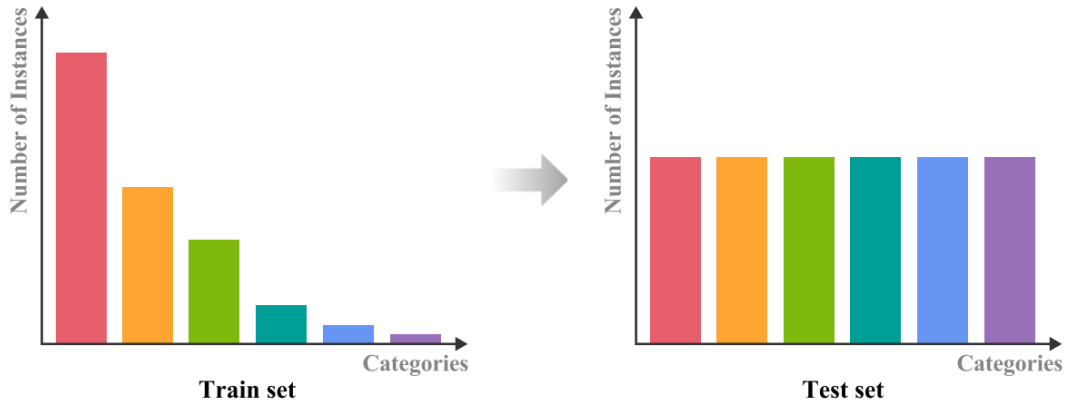


Figure 2.2: Training and Test Set Class Distribution. The figure's left side illustrates the training set's long-tailed distribution, characterized by a decreasing number of samples per class, resulting in class imbalance. The right side illustrates the balanced distribution of the test set, with each class having an equal number of samples. Figure source is [3].

In our research, we employ multiple imbalance ratios [50, 100] to generate a spectrum of class imbalance scenarios. These ratios, already defined in the introduction, represent the ratio of the number of samples in the most populated class to the least populated class, reflecting a range of real-world imbalances. This selection provides a comprehensive framework for testing and evaluating our modified AP Loss. By employing these distinct ratios, we systematically alter the number of training samples per class, creating unique challenges for the model training process and allowing a thorough assessment of performance across varied degrees of class imbalance.

It is crucial to acknowledge that although class imbalance is introduced in the training sets, the test sets of these altered datasets maintain a balanced structure. In the test data, every class possesses an equal quantity of samples, guaranteeing a fair and impartial assessment of the model's effectiveness. This method permits the evaluation of the modified AP Loss's ability to learn from imbalanced training data when applied to a balanced dataset, offering a more transparent view of its practical utility and resilience.

CHAPTER 3

METHODS

This chapter revisits the formulation of ranking-based losses, employing Oksuz et al.'s [4] notation for enhanced clarity. We focus specifically on the AP Loss [5], introducing two novel extensions designed to address the challenges of long-tailed class distributions in object detection. Our first proposed extension, 'Class Balancing for Delta Parameter' (Section 3.2), incorporates class-aware adjustments to the delta parameter within the AP Loss. The second, 'Class Balancing for Primary Term' (Section 3.3), directly modifies the primary term of the loss function. These class-balancing strategies aim to improve the model's performance on underrepresented classes.

3.1 Revisiting Ranking-based Losses

For better generalization, this chapter uses Oksuz et al.'s [4] notation and 'Identity Update' formulation to define AP Loss and make our modifications using their formulation.

Oksuz et al. [4] define the ranking based losses as:

$$\mathcal{L} = \frac{1}{Z} \sum_{i \in \mathcal{P} \cup \mathcal{N}} \left(\ell(i) - \ell^*(i) \right). \tag{3.1}$$

In this definiton, Z represents a normalization constant tailored to the specific problem at hand, $\mathcal{P}$ denotes the positives and $\mathcal{N}$ denotes the negatives. $\ell(i)$ calculates the error for each individual positive example. Thus, $\ell(i)$ is the current ranking error and $\ell^*(i)$ is the target ranking error.

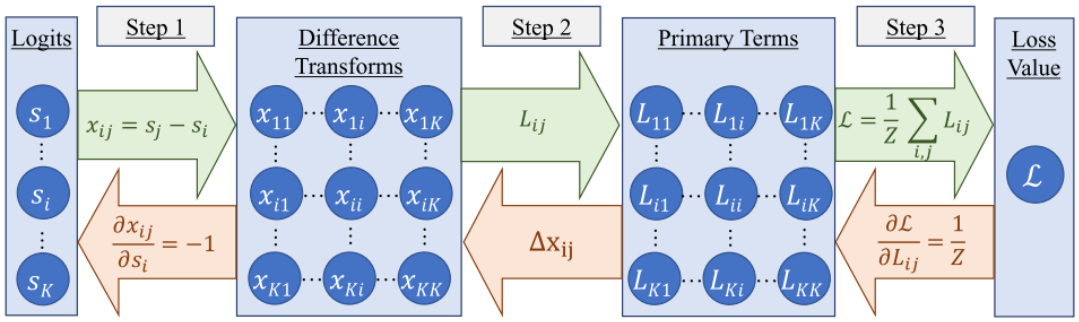


Figure 3.1: Ranking based losses computation (rightward arrows) formulation by Oksuz et al. [4]. The leftward arrows show the optimization part. Figure source is [4].

Step 1: First, calculate the difference between the scores as:

$$x_{ij} = s_j - s_i. \tag{3.2}$$

Step 2: Calculate the primary terms (L_{ij}) using the difference transforms (x_{ij}) from the previous step:

$$L_{ij} = \begin{cases} (\ell(i) - \ell^*(i))p(j \mid i), & \text{for } i \in \mathcal{P}, j \in \mathcal{N} \\ 0, & \text{otherwise.} \end{cases} \tag{3.3}$$

Since the desired ranking error $\ell^*(i)$ is zero, the primary term can be defined as:

$$L_{ij} = \begin{cases} \ell(i)p(j \mid i), & \text{for } i \in \mathcal{P}, j \in \mathcal{N} \\ 0, & \text{otherwise.} \end{cases} \tag{3.4}$$

$p(j \mid i)$ is a function that assigns a probability to each negative example $j \in \mathcal{N}$, indicating how responsible it is for the error ($\ell(i)$) of a positive example $i \in \mathcal{P}$. This makes it a probability mass function.

We need other ranking-based functions to define $\ell(i)$ and $p(j \mid i)$. First we need to establish a pairwise ranking relation between the scores s_i and s_j. A step function ($\mathrm{H}(x)$) which takes the difference calculated in step 1 (Equation 3.2) is suitable for

this purpose. This step funciton can be a piecewise linear step function or simply the unit step function. Here is how it is defined by by Chen et al. [5]:

$$H(x) = \begin{cases} 0, & x < -\delta, \\ \frac{x}{2\delta} + 0.5, & -\delta \leq x \leq \delta, \\ 1, & \delta < x. \end{cases} \tag{3.5}$$

We will build other functions using above mentioned ranking relation. Rank of the i'th example is defined by:

$$\mathrm{rank}(i) = \sum_{j \in \mathcal{P} \cup \mathcal{N}} \mathrm{H}\left(x_{ij}\right). \tag{3.6}$$

Rank of the i'th example is:

$$\mathrm{rank}^{+}(i) = \sum_{j \in \mathcal{P}} \mathrm{H}\left(x_{ij}\right). \tag{3.7}$$

We observe that $\mathrm{rank}(i)$ and $\mathrm{rank}^{+}(i)$ simply count the number of samples and positive samples respectively. Similarly we can define number of false positives:

$$\mathrm{N}_{\mathrm{FP}}(i) = \sum_{j \in \mathcal{N}} \mathrm{H}\left(x_{ij}\right). \tag{3.8}$$

Using above functions, we can define ranking-based error of sample i:

$$\ell(i) = \frac{\mathrm{N}_{\mathrm{FP}}(i)}{\mathrm{rank}(i)}, \tag{3.9}$$

and probability mass function (pmf) that distributes (i) over j as:

$$p(j \mid i) = \frac{\mathrm{H}\left(x_{ij}\right)}{\mathrm{N}_{\mathrm{FP}}(i)}. \tag{3.10}$$

Step 3: As the last step, $\mathcal{L}$ is calculated using primary terms by definition:

$$\mathcal{L} = \frac{1}{Z} \sum_{i \in \mathcal{P}} \ell(i) = \frac{1}{Z} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} L_{ij}. \tag{3.11}$$

The goal of optimizing the loss function is to calculate the gradients of the loss $\mathcal{L}$ with respect to the predicted scores s_i, denoted as $\frac{\partial \mathcal{L}}{\partial s_i}$. Once these gradients are obtained, they can be used to update the model parameters through backpropagation:

$$\frac{\partial \mathcal{L}}{\partial s_i} = \sum_{j,k} \frac{\partial \mathcal{L}}{\partial L_{jk}} \Delta x_{jk} \frac{\partial x_{jk}}{\partial s_i}. \tag{3.12}$$

Above equation breaks down the gradient calculation into three parts. Step 3; $\frac{\partial \mathcal{L}}{\partial L_{jk}}$ and step 1; $\frac{\partial x_{jk}}{\partial s_i}$ are straightforward gradients. Step2; the term Δx_{jk} replaces the differentiation of primary term L_{jk} with respect to the difference trasform x_{jk}.

For Equation 3.12, $\frac{\partial \mathcal{L}}{\partial L_{jk}}$ is $\frac{1}{|\mathcal{P}|}$ and partial derivative $\frac{\partial x_{jk}}{\partial s_i}$ has 3 cases depending on i:

$$\frac{\partial x_{jk}}{\partial s_i} = \begin{cases} \frac{\partial (s_k - s_{j=i})}{\partial s_i} = -1, & \text{if } i = j, \\ \frac{\partial (s_{k=i} - s_j)}{\partial s_i} = 1, & \text{if } i = k, \\ \frac{\partial (s_k - s_j)}{\partial s_i} = 0, & \text{if } i \neq j \text{ and } i \neq k. \end{cases} \tag{3.13}$$

Therefore, Equation 3.12 simply becomes:

$$\frac{\partial \mathcal{L}}{\partial s_i} = \frac{1}{|\mathcal{P}|} \left(\sum_j \Delta x_{ji} - \sum_j \Delta x_{ij} \right). \tag{3.14}$$

The only non-differentiable step in gradients calculation is Step 2 since the primary term L_{ij} is not differentiable with respect to the difference transform x_{ij}. To overcome this issue, in their design of AP Loss, Chen et al. borrow a fundamental concept from a classic machine learning algorithm called the perceptron. In the perceptron, the way a model learns is 'error-driven' meaning it adjusts its predictions based on the difference between what it should have predicted and what it actually predicted. Similarly, AP Loss updates its predictions based on the difference between the desired ranking of examples and the current ranking produced by the model. This error-driven approach allows the model to continuously improve how it prioritizes different detections. Incorporating this concept, the update term Δx_{ij} is defined as:

$$\Delta x_{ij} = -(L^*_{ij} - L_{ij}), \tag{3.15}$$

where L^*_{ij} represents the target primary term, which is the desired error for the pair (i, j). Ideally, L^*_{ij} should be zero, meaning that there should be no error between the

desired and actual ranking of examples. When $L^*_{ij} = 0$, the update term simplifies to:

$$\Delta x_{ij} = -(0 - L_{ij}) = L_{ij}. \tag{3.16}$$

In this case, the update term Δx_{ij} becomes equal to the primary term L_{ij} itself. This means that the model's predictions are directly adjusted based on the current errors in the ranking of examples, allowing for a more straightforward optimization of the ranking objective. Hence we can represent the gradient in Equation 3.14 in terms of primary term:

$$\frac{\partial \mathcal{L}}{\partial s_i} = \frac{1}{|\mathcal{P}|} \left(\sum_{j \in \mathcal{P} \cup \mathcal{N}} L_{ji} - \sum_{j \in \mathcal{P} \cup \mathcal{N}} L_{ij} \right). \tag{3.17}$$

We can further simplify the gradients by using the definition of primary term (Equation 3.4). There are two cases for i. If $i \in \mathcal{P}$ the gradient is:

$$\frac{\partial \mathcal{L}}{\partial s_i} = \frac{1}{|\mathcal{P}|} \left(\cancelto{0}{\sum_{j \notin \mathcal{P}} L_{ji}} + \cancelto{0}{\sum_{j \notin \mathcal{N}} L_{ji}} - \cancelto{0}{\sum_{j \notin \mathcal{P}} L_{ij}} - \sum_{j \in \mathcal{N}} L_{ij} \right). \tag{3.18}$$

If $i \in \mathcal{N}$ the gradient is:

$$\frac{\partial \mathcal{L}}{\partial s_i} = \frac{1}{|\mathcal{P}|} \left(\sum_{j \in \mathcal{P}} L_{ji} + \cancelto{0}{\sum_{j \notin \mathcal{N}} L_{ji}} - \cancelto{0}{\sum_{j \notin \mathcal{P}} L_{ij}} - \cancelto{0}{\sum_{j \notin \mathcal{N}} L_{ij}} \right). \tag{3.19}$$

Finally the gradient is:

$$\frac{\partial \mathcal{L}}{\partial s_i} = \frac{1}{|\mathcal{P}|} \left(\sum_{j \in \mathcal{P}} L_{ji} - \sum_{j \in \mathcal{N}} L_{ij} \right). \tag{3.20}$$

3.2 Proposed Extension 1: Class Balancing for Delta Parameter

Our approach is inspired by methods used to adjust loss values when data isn't evenly distributed across different categories. The main idea is to give more importance to the under-represented classes by increasing their weight in the loss calculation, a method known as class-based reweighting. By doing this, we ensure that the model pays more attention to these classes, despite their smaller size. In order to give more

importance to the under-represented classes we adjusted delta parameter. Figure 3.2 shows how piecewise linear function from Equation 3.5 changes for different δ. Our initial hypothesis suggested that a larger δ would lead to a more precise representation of the input data. Thus we implemented higher δ for under-represented clasesses and lower δ for over-represented classes. Our experiments confirmed that this approach was effective, supporting our initial assumption.

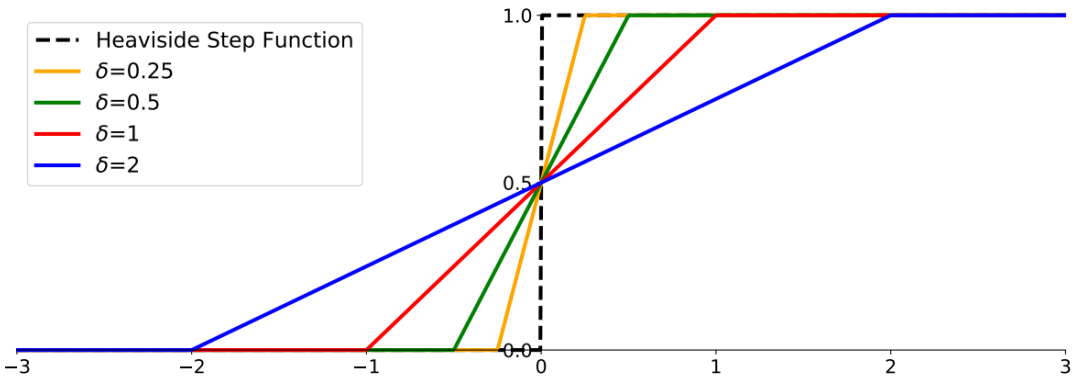


Figure 3.2: Different values of piecewise linear function (Equation 3.5) for different δ. Figure source is [5].

For adjusting the weights of each class we use Cui et al.'s [2] class weighting strategy. Their strategy is solely based on cardinality sorted class index. In below equation, β is the hyperparameter which usually takes the values in $\{0.9, 0.99, 0.999, 0.9999\}$, $n_c(i)$ is the class index of the i'th sample, and $w_c(i)$ is the final weight the be multiplied with target parameter/loss.

$$w_c(i) = \frac{1-\beta}{1-\beta^{n_c(i)}} \tag{3.21}$$

For incorporating the weight into δ we simply multiplied δ with the weight $w_c(i)$:

$$\delta_w(i) = w_c(i) \cdot \delta \tag{3.22}$$

3.3 Proposed Extension 2: Class Balancing for Primary Term

For class balancing the primary term of the loss function, we used the same weights (Equation 3.21) we used for δ. Again we simply multiplied the primary term (Equa-

tion 3.4) with weight, and the primary term becomes:

$$L_{ij}^{w} = w_c(i) L_{ij}. \tag{3.23}$$

The formulation of the primary term is updated as:

$$L_{ij}^{w} = \begin{cases} w_c(i)\ell(i)p(j \mid i), & \text{for } i \in \mathcal{P}, j \in \mathcal{N} \\ 0, & \text{otherwise.} \end{cases} \tag{3.24}$$

Accordingly the loss formula (Equation 3.11) is updated as:

$$\mathcal{L} = \frac{1}{Z} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} L_{ij}^{w} = \frac{1}{Z} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} w_c(i) L_{ij}. \tag{3.25}$$

For the gradients, we will incorporate weighted primary term into gradient calculation (Equation 3.12):

$$\frac{\partial \mathcal{L}}{\partial s_i} = \sum_{j,k} \frac{\partial \mathcal{L}}{\partial L_{wjk}} \Delta x_{jk} \frac{\partial x_{jk}}{\partial s_i}. \tag{3.26}$$

The 3rd term $\frac{\partial x_{jk}}{\partial s_i}$ is unaffected of the primary term. The first term $\frac{\partial \mathcal{L}}{\partial L_{wjk}}$ is again $\frac{1}{|\mathcal{P}|}$. The second term, which is update term Δx_{jk} will be calculated as:

$$\Delta x_{ij} = -(L_{ij}^{w*} - L_{ij}^{w}) = -(0 - L_{ij}^{w}) = L_{ij}^{w}. \tag{3.27}$$

Accordingly, as in equation (Equation 3.20) the gradients will be:

$$\frac{\partial \mathcal{L}}{\partial s_i} = \frac{1}{|\mathcal{P}|} \left(\sum_{j \in \mathcal{P}} L_{ji}^{w} - \sum_{j \in \mathcal{N}} L_{ij}^{w} \right) = \frac{1}{|\mathcal{P}|} \left(\sum_{j \in \mathcal{P}} w_c(j) L_{ji} - \sum_{j \in \mathcal{N}} w_c(i) L_{ij} \right). \tag{3.28}$$

CHAPTER 4

EXPERIMENTS

4.1 Training and Implementation Details

We used code base of Bag of Tricks repository of Zhang et al. [42] to conduct the tests. We incorporated the AP Loss implementation by Oksuz et al. [4] to carry out our tests.

We conducted experiments on the CIFAR-10-LT and CIFAR-100-LT [2] dataset using a residual network [43] with 32 layers, commonly referred to as ResNet32. For the optimizer, we employed Stochastic Gradient Descent (SGD) with a momentum of 0.9 and weight decay of 0.0002. The learning rate and weight decay were specifically tuned for each imbalance ratio and for each loss function. We performed 200 training epochs with a batch size of 128 on a single GPU, implementing a learning rate strategy that included a 5-epoch warm-up and a multi-step scheduler with a 0.01 multiplier at epochs 160 and 180. The results are presented as the average of 3 experiments with standard deviation.

4.2 Experiments on Proposed Extension 1: Modifying Delta Parameter

In order to better understand the effect of weight on δ parameter we tried both multiplying the delta and dividing it by the weight. Tables 4.1, 4.2, 4.3, 4.4 show the results for different imbalance ratios.

For Table 4.1, the best performance on average is achieved when delta is multiplied. 'Multiply' version means higher delta for head classes and lower delta for tail classes.

Table 4.1: CIFAR-10 Imbalance Ratio 50 accuracies for delta multiplication and division. Best results are shown in bold and the second best results are shown with *.

	Average	Head	Body	Tail
Baseline (Cross Entropy)	$76.69 \pm 0.43^*$	$89.03 \pm 0.21^*$	$74.31 \pm 0.87^*$	62.6 ± 0.93
AP Loss [5]	76.65 ± 0.28	88.75 ± 0.72	73.84 ± 0.60	$63.31 \pm 1.61^*$
AP Loss CB Delta Multiply [Ours]	$\mathbf{78.79 \pm 0.40}$	86.37 ± 0.75	$\mathbf{75.00 \pm 0.81}$	$\mathbf{72.48 \pm 0.17}$
AP Loss CB Delta Divide [Ours]	75.18 ± 0.30	$\mathbf{89.14 \pm 0.12}$	71.91 ± 2.14	59.84 ± 1.72

When compared with original AP Loss, 'Multiply' version declines head classes and improves tail classes. Adjunctly, 'Divide' version improves head classes while declines tail classes compared to original AP Loss. For the 'Multiply' version, since the decline of head classes is much lower compared to the huge improvement of tail classes we observe substantial improvement on average accuracy of all classes.

Table 4.2: CIFAR-10 Imbalance Ratio 100 accuracies for delta multiplication and division. Best results are shown in bold and the second best results are shown with *.

	Average	Head	Body	Tail
Baseline (Cross Entropy)	70.6 ± 0.71	82.55 ± 4.66	69.68 ± 3.11	$55.6 \pm 5.90^*$
AP Loss [5]	$71.74 \pm 0.22^*$	$88.23 \pm 0.21^*$	$\mathbf{70.03 \pm 0.98}$	51.47 ± 1.37
AP Loss CB Delta Multiply [Ours]	$\mathbf{74.77 \pm 0.09}$	85.74 ± 0.58	$69.70 \pm 1.15^*$	$\mathbf{65.22 \pm 0.28}$
AP Loss CB Delta Divide [Ours]	69.33 ± 0.44	$\mathbf{88.51 \pm 0.27}$	67.43 ± 0.96	45.64 ± 0.41

For Table 4.2, we see similar results to Table 4.1 for body, tail classes and average of all classes. However this time the best body classes results are achieved by original AP Loss. Also, as the imbalance ratio increases from 50 to 100, both 'Multiply' and 'Divide' versions show their effect more in the tail classes.

For Table 4.3, results are aligned with previous experiments of Tables 4.1, 4.2 for tail and average of classes. However for the head classes, 'Divide' options is not the best this time, moreover it declines accuracy on all classes sharply. We can speculate that overfitting to a group of classes (head classes in this case), causes the model to underperform even on that group of classes.

For Table 4.4, the direction of increases and declines of accuracies in consistent with Table 4.3. Nevertheless, as the imbalance ratio increases, the magnitude of both im-

Table 4.3: CIFAR-100 Imbalance Ratio 50 accuracies for delta multiplication and division. Best results are shown in bold and the second best results are shown with *.

	Average	Head	Body	Tail
Baseline (Cross Entropy)	$44.88 \pm 0.36^*$	69.38 ± 0.16	$44.22 \pm 0.39^*$	$20.28 \pm 0.85^*$
AP Loss [5]	44.31 ± 0.57	$\mathbf{70.49 \pm 0.51}$	43.96 ± 1.05	17.68 ± 0.70
AP Loss CB Delta Multiply [Ours]	$\mathbf{48.23 \pm 0.49}$	67.52 ± 1.03	$\mathbf{45.99 \pm 0.27}$	$\mathbf{30.59 \pm 0.16}$
AP Loss CB Delta Divide [Ours]	40.68 ± 0.59	$69.66 \pm 0.21^*$	39.42 ± 0.96	12.07 ± 0.66

Table 4.4: CIFAR-100 Imbalance Ratio 100 accuracies for delta multiplication and division. Best results are shown in bold and the second best results are shown with *.

	Average	Head	Body	Tail
Baseline (Cross Entropy)	$40.79 \pm 0.29^*$	68.52 ± 1.03	$41.35 \pm 0.74^*$	$11.67 \pm 0.36^*$
AP Loss [5]	39.98 ± 0.08	$\mathbf{69.16 \pm 0.56}$	40.54 ± 0.76	9.35 ± 0.10
AP Loss CB Delta Multiply [Ours]	$\mathbf{44.10 \pm 0.18}$	66.47 ± 0.20	$\mathbf{42.81 \pm 0.44}$	$\mathbf{22.34 \pm 0.23}$
AP Loss CB Delta Divide [Ours]	36.20 ± 0.09	$68.81 \pm 0.57^*$	33.44 ± 0.60	5.37 ± 0.37

provements and declines in performance also becomes more pronounced.

Most obviously increasing δ for a specific class makes the model focus more on that class. That is why 'Multiply' version gets best results for tail classes over all datasets and 'Divide' version gets best results for head classes on most datasets. While increasing δ improves performance on tail classes, it leads to a slight decrease in accuracy for head classes, representing a trade-off in the model's focus. This ablation study demonstrates how we can selectively increase the accuracy of specific classes by adjusting the δ parameter, providing enhanced control over the model's learning focus. We also observe that original AP Loss does is not superior to Baseline (Cross Entropy) as we expected.

4.3 Experiments on Balancing Strategy: Testing Progressive Balancing

To further enhance tail classes' performance without significantly compromising head classes' accuracy, we explored progressive weight balancing during training. This

strategy involves starting with uniform weighting and then gradually shifting the weight distribution to emphasize underrepresented classes as training progresses.

For the scheduling of the weighting, we used what Kang et al. [26] used for progressively balanced sampling, where $w^P(t)$ represents the weight for a specific class at epoch t using the progressively balanced sampling approach, t is the current epoch during the training process, T is the total number of epochs in the training process, w^U is the initial or unbalanced weight for a specific class (used at the beginning of the training process when $t = 0$), and w^{CB} is the class-balanced weight for a specific class (used at the end of the training process when $t = T$). The equation calculates the weight for a specific class at epoch t by linearly interpolating between the initial unbalanced weight w^U and the class-balanced weight w^{CB}, with the interpolation factor determined by the ratio of the current epoch t to the total number of epochs T:

$$w^P(t) = \left(1 - \frac{t}{T}\right) w^U + \frac{t}{T} w^{CB}. \tag{4.1}$$

Tables 4.5, 4.6, 4.7, 4.8 are for comparing the progressive balancing results.

Table 4.5: CIFAR-10 Imbalance Ratio 50 accuracies for comparing progressive balancing. Best results are shown in bold and the second best results are shown with *.

	Average	Head	Body	Tail
Baseline (Cross Entropy)	76.69 ± 0.43	**89.03 ± 0.21**	74.31 ± 0.87	62.6 ± 0.93
AP Loss [5]	76.65 ± 0.28	88.75 ± 0.72*	73.84 ± 0.60	63.31 ± 1.61
AP Loss CB Delta [Ours]	78.79 ± 0.40*	86.37 ± 0.75	75.00 ± 0.81*	**72.48 ± 0.17**
AP Loss CB Delta Progressive [Ours]	**79.26 ± 0.15**	87.45 ± 0.22	**75.98 ± 0.52**	71.62 ± 0.97*

Table 4.6: CIFAR-10 Imbalance Ratio 100 accuracies for comparing progressive balancing. Best results are shown in bold and the second best results are shown with *.

Method	Average	Head	Body	Tail
(Cross Entropy)	70.6 ± 0.71	82.55 ± 4.66	69.68 ± 3.11	55.6 ± 5.90
AP Loss [5]	71.74 ± 0.22	**88.23 ± 0.21**	70.03 ± 0.98*	51.47 ± 1.37
AP Loss CB Delta [Ours]	74.77 ± 0.09*	85.74 ± 0.58	69.7 ± 1.15	**65.22 ± 0.28**
AP Loss CB Delta Progressive [Ours]	**74.81 ± 0.47**	86.58 ± 0.08*	**70.98 ± 0.84**	62.94 ± 0.79*

Our initial goal was to enhance overall performance by boosting the results for head and body classes without adversely affecting the tail classes. While we observed

an increase in overall performance as evidenced in the results from CIFAR-10-LT in Tables 4.5 and 4.6, the improvement was notably modest. The enhancements in the average, head, and body classes are further evidenced by their reduced standard deviations, indicating a more consistent performance.

Table 4.7: CIFAR-100 Imbalance Ratio 50 accuracies for comparing progressive balancing. Best results are shown in bold and the second best results are shown with *.

	Average	Head	Body	Tail
Baseline (Cross Entropy)	44.88 ± 0.36	$69.38 \pm 0.16^{*}$	44.22 ± 0.39	20.28 ± 0.85
AP Loss [5]	44.31 ± 0.57	$\mathbf{70.49 \pm 0.51}$	43.96 ± 1.05	17.68 ± 0.70
AP Loss CB Delta [Ours]	$\mathbf{48.23 \pm 0.49}$	67.52 ± 1.03	$\mathbf{45.99 \pm 0.27}$	$\mathbf{30.59 \pm 0.16}$
AP Loss CB Delta Progressive [Ours]	$48.16 \pm 0.77^{*}$	69.23 ± 1.20	$45.39 \pm 0.24^{*}$	$29.20 \pm 1.53^{*}$

Table 4.8: CIFAR-100 Imbalance Ratio 100 accuracies for comparing progressive balancing. Best results are shown in bold and the second best results are shown with *.

	Average	Head	Body	Tail
Baseline (Cross Entropy)	40.79 ± 0.29	$68.52 \pm 1.03^{*}$	41.35 ± 0.74	11.67 ± 0.36
AP Loss [5]	39.98 ± 0.08	$\mathbf{69.16 \pm 0.56}$	40.54 ± 0.76	9.35 ± 0.10
AP Loss CB Delta [Ours]	$\mathbf{44.10 \pm 0.18}$	66.47 ± 0.20	$\mathbf{42.81 \pm 0.44}$	$\mathbf{22.34 \pm 0.23}$
AP Loss CB Delta Progressive [Ours]	$43.65 \pm 0.55^{*}$	67.22 ± 0.62	$42.18 \pm 0.55^{*}$	$20.83 \pm 0.82^{*}$

For the CIFAR-100-LT experiments presented in Tables 4.7 and 4.8, the results diverge from those observed in CIFAR-10-LT (Tables 4.5 and 4.6). In the CIFAR-100-LT experiments, performance decreases across all class groups, and standard deviations increase, suggesting less consistency in model performance.

Progressive balancing does not consistently yield improvements; and when improvements do occur, they are marginal. Consequently, we conclude that progressive balancing is ineffective in this context and have decided to discontinue its use in further experiments. Instead, we shifted our focus to core modifications of the AP Loss function.

4.4 Comparison with Other Methods

Our experiments on CIFAR-10 and CIFAR-100 with imbalance ratios 50 and 100 substantiate the effectiveness of class-aware adjustments within the AP Loss framework for mitigating long-tailed dataset biases. Default AP Loss is on par with baseline (cross-entropy) results. However our modifications, namely AP Loss with Class Balanced Delta (Section 3.2) and AP Loss with Class Balanced Primary term (Section 3.3) significantly improve default AP Loss. The methods are indicated in the tables as 'AP Loss CB Delta' and 'AP Loss CB Primary Term' in order. We compared our methods with 4 different other methods.

Focal Loss [44] modifies the standard cross-entropy loss by introducing a scaling factor that adjusts the contribution of each example to the loss based on how well the model classifies it. Specifically, it incorporates a focusing parameter, usually denoted as γ, which reduces the loss contribution from easy examples and amplifies it for misclassified or hard examples. This is achieved by scaling the standard cross-entropy loss with a factor $(1-p_t)^\gamma$, where p_t is the model's estimated probability for the class with the correct label. As a result, focal loss dynamically adjusts the emphasis on correcting misclassifications, making it particularly effective in scenarios where there is a significant class imbalance or when dealing with background noise in tasks like object detection. This strategic focus helps in fine-tuning the model's sensitivity to underrepresented classes, leading to improved learning outcomes and more robust models.

The Class-Balanced Focal Loss, proposed by Cui et al. [2], addresses class imbalance in datasets by modifying the standard Focal Loss with a weighting factor based on the effective number of samples. This method uses the same class balancing strategy(Section 3.2) with other class balanced loss functions in this thesis.

The DRO-LT (Distributionally Robust Optimization for Long-Tail) [45] loss is a novel loss function designed to enhance the learning of features in neural networks trained on long-tail data distributions. It incorporates principles from robust optimization to address the high variability and uncertainty associated with infrequently occurring classes. By creating a robustness margin around the estimated class cen-

troids, the DRO-LT loss adjusts for the potential estimation errors due to the limited number of samples in tail classes. It is formulated to consider the worst-case distribution within an uncertainty set, ensuring that the feature representations are reliable and effective even when actual class distributions deviate from those seen during training. This loss function is particularly valuable for improving the recognition accuracy of tail classes without compromising the performance on head classes, making it suitable for applications with imbalanced data distributions.

The LDAM (Label-Distribution-Aware Margin) Loss, introduced by Kaidi Cao et al., [41] is a loss function designed to address the class imbalance problem in deep learning. By integrating a margin-based generalization bound into the loss function, LDAM imposes larger decision margins on minority classes compared to majority classes, which theoretically supports better generalization for these underrepresented classes. This approach modifies the conventional soft margin loss, encouraging the model to prioritize accurate classification of minority classes without losing overall performance.

Table 4.9: CIFAR-10 Imbalance Ratio 50 accuracies for comparison with other methods. The results with '†' are retrieved from Baltaci et al. [1]. Best results are show with bold and the second best results are shown with '*'.

	Average	Head	Body	Tail
Baseline (Cross Entropy)	76.69 ± 0.43	**89.03** ± 0.21	74.31 ± 0.87	62.6 ± 0.93
Focal [44]	76.57 ± 1.44	88.41 ± 0.49*	74.78 ± 2.09	62.56 ± 2.15
Focal CB† [2]	79.55 ± 0.27	NA	NA	NA
DRO-LT† [45]	**85.65 ± 0.19**	NA	NA	NA
LDAM [41]	81.16 ± 0.35*	85.35 ± 0.61	**78.8 ± 1.01**	**77.93 ± 0.58**
AP Loss [5]	76.65 ± 0.28	88.75 ± 0.72	73.84 ± 0.60	63.31 ± 1.61
AP Loss CB Delta [Ours]	78.79 ± 0.40	86.37 ± 0.75	75.00 ± 0.81	72.48 ± 0.17*
AP Loss CB Primary Term [Ours]	77.99 ± 0.34	87.62 ± 0.43	75.36 ± 0.76*	67.80 ± 0.24

In examining Table 4.9, which presents CIFAR-10 data with an imbalance ratio of 50, we observe that DRO-LT method outperforms others with the highest average accuracy of 85.65%, highlighting its robustness across the dataset. However, it lacks specific data for the Head, Body, and Tail categories, since the results are retrieved form Baltaci et al. [1]. LDAM is particularly effective in the Tail category, achieving

the highest accuracy of 77.93%, which indicates its strength in addressing underrepresented classes. Among our proposed methods, AP Loss CB Delta demonstrated substantial efficacy, particularly in the Tail category, with a second-best accuracy of 72.48%. Conversely, the AP Loss CB Primary Term variant, while slightly less effective overall with an average accuracy of 77.99%, showed notable performance in the Body category, which may indicate its utility in improving learning outcomes for moderately represented classes.

Table 4.10: CIFAR-10 Imbalance Ratio 100 accuracies for comparison with other methods. The results with '†' are retrieved from Baltaci et al. [1]. Best results are show with bold and the second best results are shown with '*'.

	Average	Head	Body	Tail
Baseline (Cross Entropy)	70.6 ± 0.71	82.55 ± 4.66	69.68 ± 3.11	55.6 ± 5.90
Focal [44]	71.39 ± 0.76	**88.28 ± 0.32**	69.99 ± 0.45	50.26 ± 2.43
Focal CB† [2]	73.92 ± 0.34	NA	NA	NA
DRO-LT† [45]	**82.78 ± 0.12**	NA	NA	NA
LDAM [41]	78.19 ± 0.12*	81.81 ± 0.70	**75.89 ± 0.88**	**75.65 ± 1.32**
AP Loss [5]	71.74 ± 0.22	88.23 ± 0.21*	70.03 ± 0.98	51.47 ± 1.37
AP Loss CB Delta [Ours]	74.77 ± 0.09	85.74 ± 0.58	69.7 ± 1.15	65.22 ± 0.28*
AP Loss CB Primary Term [Ours]	72.58 ± 0.57	85.08 ± 0.10	70.27 ± 0.36*	58.23 ± 1.62

In the analysis of the CIFAR-100 dataset with an imbalance ratio of 50 in Table 4.10, the performance of various loss functions continues to align closely with observations from previous CIFAR-10 evaluations. The table highlights DRO-LT as the top performer with the highest average accuracy of 82.78%, reaffirming its effectiveness across different datasets and imbalance settings. LDAM also maintains strong performance, particularly in the Body and Tail classes where it achieves the highest accuracies, signaling its capability to effectively manage underrepresented classes. Our methods, AP Loss CB Delta and AP Loss CB Primary Term, exhibit varied results; AP Loss CB Delta notably secures the second-best average accuracy of 74.77% and performs notably in the Tail classes.

The performance data from the CIFAR-100 dataset with an imbalance ratio of 50 in Table 4.11 showcases the efficacy of various loss functions. DRO-LT stands out with the highest average accuracy at 53.6%, indicating its superior handling of imbalanced

Table 4.11: CIFAR-100 Imbalance Ratio 50 accuracies for comparison with other methods. The results with '†' are retrieved from Baltaci et al. [1]. Best results are show with bold and the second best results are shown with '*'.

	Average	Head	Body	Tail
Baseline (Cross Entropy)	44.88 ± 0.36	69.38 ± 0.16*	44.22 ± 0.39	20.28 ± 0.85
Focal [44]	45.24 ± 0.26	**70.77 ± 0.56**	44.45 ± 1.00	19.72 ± 0.44
Focal CB† [2]	42.91 ± 0.55	NA	NA	NA
DRO-LT† [45]	**53.6 ± 0.10**	NA	NA	NA
LDAM [41]	47.41 ± 0.37	64.49 ± 0.41	**45.99 ± 0.86**	**31.24 ± 0.45**
AP Loss [5]	44.31 ± 0.57	70.49 ± 0.51	43.96 ± 1.05	17.68 ± 0.70
AP Loss CB Delta [Ours]	48.23 ± 0.49*	67.52 ± 1.03	**45.99 ± 0.27**	30.59 ± 0.16*
AP Loss CB Primary Term [Ours]	42.59 ± 0.32	58.99 ± 0.38	44.58 ± 0.73	23.69 ± 0.47

datasets. But this time, for CIFAR-100; our AP Loss CB Delta method also performs commendably, securing the second-best average accuracy at 48.23% and showing notable improvement in the Tail classes. Also its gap with LDAM on tail classes significantly narrows.

Table 4.12: CIFAR-100 Imbalance Ratio 100 accuracies for comparison with other methods. The results with '†' are retrieved from Baltaci et al. [1]. Best results are show with bold and the second best results are shown with '*'.

	Average	Head	Body	Tail
Baseline (Cross Entropy)	40.79 ± 0.29	68.52 ± 1.03	41.35 ± 0.74	11.67 ± 0.36
Focal [44]	40.36 ± 0.67	**69.66 ± 0.60**	40.05 ± 1.04	10.48 ± 0.55
Focal CB† [2]	40.1 ± 0.19	NA	NA	NA
DRO-LT† [45]	**48.25 ± 0.15**	NA	NA	NA
LDAM [41]	44.91 ± 0.47*	62.91 ± 0.47	**45.34 ± 1.09**	**25.93 ± 0.41**
AP Loss [5]	39.98 ± 0.08	69.16 ± 0.56*	40.54 ± 0.76	9.35 ± 0.10
AP Loss CB Delta [Ours]	44.10 ± 0.18	66.47 ± 0.20	42.81 ± 0.44*	22.34 ± 0.23*
AP Loss CB Primary Term [Ours]	34.62 ± 0.65	51.99 ± 2.14	37.07 ± 0.75	14.28 ± 0.23

The CIFAR-100 dataset with an imbalance ratio of 100 demonstrates significant performance differences among various loss functions, as shown in the table. DRO-LT leads with the highest average accuracy of 48.25%. The AP Loss CB Delta variant from our methods also performs notably, achieving strong results in the Tail segment,

with a second-best score of 22.34%.

The experimental results from our studies on CIFAR-10 and CIFAR-100 datasets, with imbalance ratios of 50 and 100, firmly validate the efficacy of class-aware modifications within the AP Loss framework for addressing biases in long-tailed datasets. While the default AP Loss performs comparably to the baseline cross-entropy results, our enhancements, specifically AP Loss with Class Balanced Delta exhibit significant improvements. The DRO-LT method's consistent outperformance across tables suggests it as a potent solution for dealing with class imbalances, warranting further exploration and potential adaptation in future works. These modifications have proven effective in enhancing the model's accuracy, particularly in the underrepresented classes, demonstrating considerable gains over traditional methods such as Focal Loss, Class-Balanced Focal Loss, DRO-LT, and LDAM. Our experimental results underscore the importance of tailored loss functions in mitigating the challenges posed by class imbalance, thereby improving the robustness and fairness of learning algorithms in handling diverse and skewed datasets.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

In this thesis, we explored the challenge of long-tailed class distributions in visual recognition and proposed two novel extensions to the AP Loss to address this issue. Our first extension, AP Loss with Class Balancing for Delta Parameter (Section 3.2), incorporated class-aware adjustments to the delta parameter within the AP Loss. The second extension, Class Balancing for Primary Term (Section 3.3), directly modified the primary term of the loss function.

Our ablation studies (Sections 4.2, 4.3) provided insights into the behavior of the delta parameter and the impact of progressive balancing. We observed that increasing the delta for a specific class makes the model focus more on that class, leading to improved accuracy on tail classes at the cost of slight performance drops on head classes. However, we found that progressive balancing did not yield consistent improvements, and decided to focus on core modifications to the AP Loss.

The comprehensive experiments (Section 4) conducted as part of this thesis on the CIFAR-10 and CIFAR-100 datasets with imbalance ratios of 50 and 100 have yielded significant insights into the effectiveness of class-aware adjustments within the AP Loss framework. These studies demonstrate that while the default AP Loss aligns closely with the performance of the baseline (cross-entropy loss), our specialized extensions AP Loss with Class Balanced Delta (Section 3.2) and AP Loss with Class Balanced Primary Term (Section 3.3) provide substantial improvements, particularly in handling imbalances within the datasets.

In our comparative analysis, we benchmarked our proposed methods against several state-of-the-art approaches designed to address long-tailed class distributions, including Focal Loss [44], Class-Balanced Focal Loss [2], DRO-LT [45], and LDAM [41]. While DRO-LT demonstrated the highest average accuracy in most settings, our approach showed substantial improvements in the accuracy of tail classes, which are the most challenging to learn in long-tailed scenarios. Notably, on the CIFAR-100-LT dataset with an imbalance ratio of 100, our method achieved the second-best performance on tail classes, surpassing Focal Loss, Class-Balanced Focal Loss, and LDAM.

In summary, our work demonstrates the effectiveness of class-aware modifications to the AP Loss framework for mitigating the challenges posed by long-tailed visual recognition datasets. By focusing on the core aspects of the loss function and incorporating class-specific information, we have shown significant improvements in the model's ability to recognize underrepresented classes without compromising overall performance.

5.2 Future Work

Based on the findings and insights gained from this thesis, several potential avenues for future research can be explored:

1. Explore the applicability of our approach to other visual recognition tasks and datasets, assessing its generalizability and robustness.
2. Develop adaptive strategies for determining the optimal class-specific weights and delta adjustments based on the dataset characteristics and class distribution.

By addressing these future research directions, we can further advance the development of robust and balanced visual recognition systems that effectively handle the challenges posed by long-tailed data distributions. The insights gained from this thesis lay the foundation for future work in this critical area of computer vision and machine learning.

REFERENCES

[1] Z. S. Baltacı, "Quantifying and mitigating class imbalance in long-tailed visual recognition," Master's thesis, Middle East Technical University, 2022.

[2] 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.

[3] L. Yang, H. Jiang, Q. Song, and J. Guo, "A survey on long-tailed visual recognition," *International Journal of Computer Vision*, vol. 130, no. 7, pp. 1837–1872, 2022.

[4] 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.

[5] K. Chen, W. Lin, J. Li, J. See, J. Wang, and J. Zou, "Ap-loss for accurate one-stage object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 11, pp. 3782–3798, 2020.

[6] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural computation*, vol. 29, no. 9, pp. 2352–2449, 2017.

[7] G. Algan and I. Ulusoy, "Image classification with deep learning in the presence of noisy labels: A survey," *Knowledge-Based Systems*, vol. 215, p. 106771, 2021.

[8] W. Wang, Y. Yang, X. Wang, W. Wang, and J. Li, "Development of convolutional neural network and its application in image classification: a survey," *Optical Engineering*, vol. 58, no. 4, pp. 040901–040901, 2019.

[9] Y. Wei, S. Tran, S. Xu, B. Kang, M. Springer, *et al.*, "Deep learning for retail

product recognition: Challenges and techniques," *Computational intelligence and neuroscience*, vol. 2020, 2020.

[10] Y. Bazi, L. Bashmal, M. M. A. Rahhal, R. A. Dayil, and N. A. Ajlan, "Vision transformers for remote sensing image classification," *Remote Sensing*, vol. 13, no. 3, p. 516, 2021.

[11] P. K. Mallick, S. H. Ryu, S. K. Satapathy, S. Mishra, G. N. Nguyen, and P. Tiwari, "Brain mri image classification for cancer detection using deep wavelet autoencoder-based deep neural network," *IEEE Access*, vol. 7, pp. 46278–46287, 2019.

[12] T. Turay and T. Vladimirova, "Toward performing image classification and object detection with convolutional neural networks in autonomous driving systems: A survey," *IEEE Access*, vol. 10, pp. 14076–14119, 2022.

[13] 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.

[14] Y. Zhang, B. Kang, B. Hooi, S. Yan, and J. Feng, "Deep long-tailed learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[15] D. A. Dablain, C. Bellinger, B. Krawczyk, and N. Chawla, "Efficient augmentation for imbalanced deep learning," *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pp. 1433–1446, 2022.

[16] X. Zhang, Z. Wang, D. Liu, Q. Lin, and Q. Ling, "Deep adversarial data augmentation for extremely low data regimes," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, pp. 15–28, 2021.

[17] Q. Kang, X. Chen, S. Li, and M. Zhou, "A noise-filtered under-sampling scheme for imbalanced classification," *IEEE Transactions on Cybernetics*, vol. 47, pp. 4263–4274, 2017.

[18] C. Ho, D. H. Park, W. Yang, and Y. Chang, "Sequenced-replacement sampling for deep learning," *ArXiv*, vol. abs/1810.08322, 2018.

[19] J. Shi, D. Song, S. Zheng, Y. Hu, S. Chen, and F. Pei, "Bidirectional sampling method for imbalanced data," vol. 12779, pp. 127792H – 127792H–7, 2023.

[20] Y. Wang, G. Huang, S. Song, X. Pan, Y. Xia, and C. Wu, "Regularizing deep networks with semantic data augmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, pp. 3733–3748, 2020.

[21] G. Szlobodnyik and L. Farkas, "Data augmentation by guided deep interpolation," *Appl. Soft Comput.*, vol. 111, p. 107680, 2021.

[22] J. M. Johnson and T. Khoshgoftaar, "Cost-sensitive ensemble learning for highly imbalanced classification," *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1427–1434, 2022.

[23] W. Zheng and H. Zhao, "Cost-sensitive hierarchical classification via multi-scale information entropy for data with an imbalanced distribution," *Applied Intelligence*, vol. 51, pp. 5940 – 5952, 2021.

[24] K. R. M. Fernando and C. P. Tsokos, "Dynamically weighted balanced loss: Class imbalanced learning and confidence calibration of deep neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, pp. 2940–2951, 2021.

[25] M. Li, Y. ming Cheung, and Y. Lu, "Long-tailed visual recognition via gaussian clouded logit adjustment," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6919–6928, 2021.

[26] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis, "Decoupling representation and classifier for long-tailed recognition," in *International Conference on Learning Representations*, 2019.

[27] Z. Zhong, J. Cui, S. Liu, and J. Jia, "Improving calibration for long-tailed recognition," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16484–16493, 2020.

[28] T. Li, L. Wang, and G. Wu, "Self supervision to distillation for long-tailed visual recognition," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 610–619, 2021.

[29] G. Nam, S. Jang, and J. Lee, "Decoupled training for long-tailed classification with stochastic representations," *ArXiv*, vol. abs/2304.09426, 2023.

[30] W. Zheng, B. Zhang, J. Lu, and J. Zhou, "Deep relational metric learning," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12045–12054, 2021.

[31] S. Parisot, P. Esperança, S. G. McDonagh, T. Madarász, Y. Yang, and Z. Li, "Long-tail recognition via compositional knowledge transfer," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6929–6938, 2021.

[32] J. Zhang, J. Song, L. Gao, Y. Liu, and H. Shen, "Progressive meta-learning with curriculum," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, pp. 5916–5930, 2022.

[33] S. Zhang, C. Chen, X. Hu, and S. Peng, "Balanced knowledge distillation for long-tailed learning," *Neurocomputing*, vol. 527, pp. 36–46, 2021.

[34] J. Yang, X. Zhu, A. Bulat, B. Martínez, and G. Tzimiropoulos, "Knowledge distillation meets open-set semi-supervised learning," *ArXiv*, vol. abs/2205.06701, 2022.

[35] W. Wang, M. Feiszli, H. Wang, J. Malik, and D. Tran, "Open-world instance segmentation: Exploiting pseudo ground truth from learned pairwise affinity," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4412–4422, 2022.

[36] Q. Gui, X. Wu, and B. Niu, "Class-aware pseudo labeling for non-random missing labels in semi-supervised learning," *2022 IEEE Eighth International Conference on Multimedia Big Data (BigMM)*, pp. 138–143, 2022.

[37] J. He, J. Qiu, A. Zeng, Z. Yang, J. Zhai, and J. Tang, "Fastmoe: A fast mixture-of-expert training system," *ArXiv*, vol. abs/2103.13262, 2021.

[38] Y. J. Kim, A. Awan, A. Muzio, A. F. Cruz-Salinas, L. Lu, A. Hendy, S. Rajbhandari, Y. He, and H. H. Awadalla, "Scalable and efficient moe training for multitask multilingual models," *ArXiv*, vol. abs/2109.10465, 2021.

[39] D. Xu, J. Deng, and W. Li, "Revisiting ap loss for dense object detection: Adaptive ranking pair selection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14187–14196, June 2022.

[40] S. Zhang, C. Chen, and S. Peng, "Reconciling object-level and global-level objectives for long-tail detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18982–18992, 2023.

[41] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, "Learning imbalanced datasets with label-distribution-aware margin loss," *Advances in neural information processing systems*, vol. 32, 2019.

[42] Y. Zhang, X.-S. Wei, B. Zhou, and J. Wu, "Bag of tricks for long-tailed visual recognition with deep convolutional neural networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, pp. 3447–3455, 2021.

[43] 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.

[44] 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.

[45] D. Samuel and G. Chechik, "Distributional robustness loss for long-tail learning," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9495–9504, 2021.