Exploring Deep Learning Techniques for Long-Tailed Recognition: Methods, Models, and Analysis

Master's Thesis in
Electrical Engineering

By

Christine Annelise Midtgaard
au521655
Study Program: Electrical Engineering

Supervisor
Kim Bjerge
Associate Professor
kbe@ece.au.dk

M.Sc. in Electrical Engineering
Department of Engineering Science & Technology
Aarhus University

January 3, 2025

Abstract

This thesis, titled *Exploring Deep Learning Techniques for Long-Tailed Recognition: Methods, Models, and Analysis*, by Christine Annelise Midtgaard, explores the challenges and solutions related to training deep learning models on long-tailed datasets. Long-tailed datasets, where a few classes dominate with abundant samples while many classes have sparse representation, pose significant challenges for traditional training methods. These imbalances often lead to models that perform well on majority classes but struggle to recognize or generalize to minority (tail) classes.

This thesis focuses on evaluating and implementing state-of-the-art methods for long-tailed learning, as outlined in the survey Deep Long-Tailed Learning: A Survey by Zhang et al. The methodologies explored include advanced sampling strategies, re-weighted loss functions, and modifications to deep learning architectures tailored to imbalanced data.

A unique application of these methods is demonstrated on a custom dataset of moth images collected near the equator, where the goal is accurate species identification. Through a series of experiments, the thesis investigates how different approaches to long-tailed learning impact model performance across head, middle, and tail classes.

The findings contribute to understanding the efficacy of these methods and provide insights into best practices for handling real-world long-tailed datasets.

Acknowledgements

I would like to thank my supervisor, Kim Bjerge, for their guidance, constructive feedback, and inspiring conversations throughout the development of this thesis. I am also deeply thankful to my family for providing me with room for my frustrations, and for cooking meals during the challenging times of this journey. I want to thank my friend, Line, for her valuable insights in writing a thesis. A special thanks to boyfriend for his outstanding patience and understanding.

Contents

Chapter 1

Introduction

This thesis focuses on the problem with long-tailed datasets. The problem with training a deep learning model on long-tailed datasets is that the model will effectively the data from the classes with most samples, and not the classes with few samples. The finsihed model will then not recognize an input from the tail classes. Most real-world datasets follows a long-tailed structure, hence the need for a reliable method to detect examples of tail-class data. The aim of this thesis is to try out some of the methods tackling the long-tailed problem for deep learning described in the paper *Deep Long-Tailed Learning: A Survey* by Zhang et al.[1] to find a method for long-tailed learning that works on a specific long-tailed dataset of images of moths taken around equator. The goal of the moth dataset is to identify species.

1.1 Problem Definition

Define the problem formally, including key terms like "head classes" and "tail classes." Provide an example or visualization of a long-tailed dataset (optional, but impactful). Connect the problem definition to your specific application (the moth dataset).

1.1.1 Goals of this thesis

Clearly outline the goals of the thesis, emphasizing measurable or specific objectives (e.g., evaluate certain methods, optimize performance on tail classes, etc.). Mention how these goals contribute to the field of long-tailed learning.

1.1.2 Approach

Briefly summarize the approach you'll take to achieve your goals, such as implementing and comparing methods from Zhang et al.'s survey. Highlight any novel aspects of your work (e.g., unique dataset, evaluation framework, or combination of methods).

1.1.3 Scope of this thesis

Specify the scope to manage reader expectations. For example: Focus is on image classification, not other tasks like object detection. Methods are tested on a specific dataset, not generalized across all domains. The evaluation is limited to certain metrics (like F1 score) and does not cover deployment concerns.

1.2 Motivation

Deepen the discussion on why the problem is significant, including real-world implications. Mention the importance of biodiversity studies or the challenges of species identification with limited samples. Discuss broader impacts, such as how solving long-tailed learning problems can benefit other fields.

1.3 Reading Guide

Provide a roadmap for the thesis structure. Mention what each chapter will cover and how they relate to each other. Example: "Chapter 2 provides the theoretical background for long-tailed learning and deep learning methodologies. Chapter 3 describes the methodology used in this thesis, including dataset preparation and implementation details. Chapters 4 and 5 present the experimental results and analysis, followed by conclusions and suggestions for future work in Chapter 6."

1.4 Related Work

A section that describes the work related to this thesis. Maybe it needs to be a chapter following the introduction. Maybe it needs to be in the discussion.

Chapter 2

Background

This chapter presents the different background topics of the thesis work, which are the long-tailed datasets, model architectures *Convolutional Neural Networks (CNN)* and *Visual Transformers (VT)*, the deep long-tailed learning methods *Class Re-balancing (CR)*, *Information Augmentation (IA)*, and *Module Improvement (MI)*. These topics will be explained for the reader.

Maybe mention image classification, as it is the primary goal of this thesis.

2.1 Long-Tailed Dataset

A short introductory paragraph explaining why these topics are relevant and how they tie into the thesis. Contextualize the sections—e.g., "Long-tailed datasets are central to this thesis, as they represent the primary challenge. Model architectures and classic long-tailed methods form the foundation of the approaches explored in this work."

Describe the difference between class-imbalanced learning and long-tailed learning.

2.2 Model Architechtures

Briefly describe the role of deep learning models in handling long-tailed datasets.

2.2.1 Convolutional Neural Networks

Add historical context (e.g., their success in computer vision tasks). Highlight specific CNNs used in this thesis (e.g., ResNet, MobileNet).

2.2.2 Visual Transformers

Explain their advantages over CNNs for certain tasks. Mention why they are relevant for handling long-tailed datasets.

2.3 Classic Long-Tailed Methods

Introduce the three methods (CR, IA, MI) with a brief explanation of their purpose.

Following the paper *Deep Long-Tailed Learning: A Survey* [1], the existing deep long-tailed learning methods are grouped into three main categories based on their technical approach: class re-balancing, information augmentation, and module improvement. These categories are further divided onto sub-categories: re-sampling, class-sensitive learning, logit adjustment, transfer learning, data augmentation, representation learning, classifier desing, decoupled training, and ensemble learning (TODO: create a figure like figure 2 in the paper). This thesis does not aim to examine all the beforementioned method, but aims to find a deep learning approach to a specific problem. The backgrounds of the methods used in this thesis are described in this section.

2.3.1 Class Re-balancing

The class re-balancing method aims to re-balance the effect of the imbalanced training dataset, and has three main sub-categories: re-sampling, class-sensitive learning, and logit adjustment [1].

Re-sampling

The traditional way to sample when training deep networks is bases on mini-batch gradient descent with random sampling. This means that each sample has an equal probability of being sampled. When sampling from an imbalanced dataset, samples from head classes naturally occur more often, and thus have higher chance of being sampled than samples from tail classes, making the resulting deep models biased towards head classes. Re-sampling is a method that adresses this problem by adjusting the number of samples per class in each sample batch for model training.

Class-sensitive Learning

Class-sensitive learning incorporates strategies to adjust the loss function, making it more sensitive to the imbalanced nature of the dataset. This approach directly modifies the optimization process to prioritize learning from under-represented tail classes.

Loss Functions for Class-Sensitive Learning

Re-weighting is a method that attempts to adjust the training loss values for different classes by multiplying them with different weights [1].

Cross-Entropy Loss Softmax loss is the standard loss function used for classification tasks. It converts logits into probabilities via the softmax function and calculates the cross-entropy between predicted probabilities and ground-truth labels. While effective for balanced datasets, softmax loss struggles with long-tailed datasets because it assumes equal importance for all classes, leading to biased predictions favoring head classes.

Weighted Softmax Loss
Focal Loss
Class-Balanced Loss
Balanced Softmax Loss
LDAM Loss
Equalization Loss

2.3.2 Information Augmentation

Data augmentation techniques tailored for long-tailed datasets.

Transfer Learning

Data Augmentation

2.3.3 Module Improvement

Architectural changes to improve tail-class representation.

Chapter 3

Methodology

This chapter describes the methods and approaches used in the experiments. This includes dataset preparation, models, loss functions, etc.

3.1 Overview of Methodology Approach

A high-level description of the approach to tackling the long-tailed dataset problem, including an explanation of the overall strategy, such as balancing techniques, model selection, and any specific objectives that guide the methodology.

3.2 Algorithm Selection and Rationale

Description of the algorithms or model architectures chosen, such as ConvNeXt or MobileNetV2, and why they are appropriate for long-tailed learning. Discussion of the strengths and limitations of these models in addressing the challenges posed by imbalanced data.

3.3 Long-tailed Learning Techniques

Description of the specific methods used to address class imbalance, such as data sampling (e.g., oversampling/undersampling), class re-weighting, or advanced approaches like LDAM or DRW. Justification for selecting these techniques, potentially referencing prior research (e.g., from Deep Long-Tailed Learning: A Survey by Zhang et al.).

3.4 Loss Functions

Explanation of the different loss functions explored, such as cross-entropy, focal loss, LDAM loss, etc., and their relevance for long-tailed learning. Rationale for each loss function's inclusion, focusing on its expected benefits for imbalanced classes and how it addresses the bias toward majority classes.

3.5 Data Imbalance Handling Strategies

Detailed explanation of the techniques for creating and handling an imbalanced dataset, such as generating imbalanced training and test sets. Any adjustments to the data pipeline to ensure that class distributions are maintained or specifically structured, as needed for the experiments.

3.6 Evaluation Strategies

Justification for the metrics and evaluation approach, such as using weighted or macro F1 scores. Explanation of how you plan to assess performance across different class groups (e.g., head, middle, tail) to capture the model's performance on minority classes.

3.7 implementation Details

Brief technical explanations of any unique or customized methods implemented in code, especially if they differ from standard practices. Examples of changes made to existing algorithms or functions to adapt them for the long-tailed learning problem.

Chapter 4

Experimental Setup

This chapter focuses on the on the implementation details of the experiments conducted in this thesis. Here, the specifics of the training configurations are described.

4.1 Dataset Specifications

Details about the dataset(s) used, including size, source, and preprocessing steps. Description of class imbalance characteristics (if applicable), and the train/validation/test splits.

4.2 Data Preprocessing

Any transformations, augmentations, or normalization applied to the dataset before feeding it to the model. Information on how you handled class imbalance (e.g., re-sampling techniques or synthetic data generation).

4.3 Model Architecture Settings

Description of the model(s) used, including any specific architecture choices, hyperparameters, or modifications. Brief details on why these models were chosen, especially if you're comparing multiple models.

4.4 Training Configurations

Hyperparameters, such as batch size, learning rate, optimizer type (like Adam or SGD), and regularization techniques (e.g., dropout, weight decay). Any specific settings for handling long-tailed data, such as dynamic re-weighting, if applicable.

4.5 Evaluation Metrics

Explanation of the metrics used to assess model performance, such as accuracy, F1 score, precision, recall, or custom metrics for imbalanced datasets. Justification for choosing each metric, especially if they help address challenges with imbalanced data.

4.6 Hardware and Software Configurations

Hardware details (e.g., number of GPUs, CPU type, memory, etc.). Software environment, including the versions of libraries and frameworks (e.g., PyTorch, TensorFlow) used.

4.7 Reproducibility Considerations

Steps taken to ensure that results can be reproduced, such as random seed initialization and details on dataset versions. Any scripts, configurations, or instructions for reproducing experiments.

Chapter 5

Results and Analysis

Presentation of your findings, with tables, charts, and explanations for each tested method's performance.
Brief overview of the chapter's purpose. Recap the evaluation goals (e.g., assessing model performance across head, middle, and tail classes, and comparing methods).

5.1 Overall Results

Present the aggregate performance of all tested models and methods. Use tables or charts to summarize key results (e.g., overall accuracy, F1 scores). Highlight trends or notable observations across the methods.

5.2 Head, Middle, and Tail Class Performance

Break down the performance into head, middle, and tail class groups. Include visualizations (e.g., bar plots or line graphs) showing metrics like F1 score or accuracy for each group. Discuss how well the methods balance performance across these groups, particularly focusing on tail classes.

5.3 Comparison of Loss Functions

Analyze how different loss functions (e.g., cross-entropy, focal loss, LDAM) impact performance. Use visual aids to compare results (e.g., per-class performance or confusion matrices). Discuss trade-offs, strengths, and weaknesses of each loss function.

5.4 Qualitative Results

Optional.
Provide examples of correctly and incorrectly classified samples, especially for tail classes. Include visualizations or images of difficult cases to highlight challenges in tail-class prediction.

5.5 Ablation Studies

Optional.
If applicable, evaluate the impact of individual components or configurations (e.g., re-weighting, sampling). Discuss how removing or altering specific aspects affects performance.

5.6 Summary and Discussion

Recap the key findings, such as which methods or loss functions performed best and why. Connect these findings to the thesis objectives and broader implications for long-tailed learning.

Chapter 6

Conclusion and Future Work

Summary of the work, contributions, and suggestions for future improvements or research directions.

6.1 Revisiting the Goals of the Thesis

6.2 Future Work

Bibliography

[1] Yifan Zhang et al. “Deep long-tailed learning: A survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

Appendix A

Results

Tables of the results from training.