Convergent Learning for Class Imbalance: A Unified Approach to Long-Tail Recognition in Image Classification

A Major Qualifying Project (MQP) Report
Submitted to the Faculty of
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements
for the Degree of Bachelor of Science in

Data Science

By:

Bishoy Soliman Hanna

Project Advisors:

Professor: Ziming Zhang

Date: Feb 2024

Abstract

Addressing the complex challenge of long-tail recognition (LTR) in image classification, this study introduces an innovative integrated approach that meticulously combines cross-entropy[1], contrastive learning[2], and imbalanced class loss functions. The prevalence of class imbalances within datasets significantly hampers the effectiveness of conventional machine learning models by skewing performance toward majority classes and neglecting the minority ones. To combat this issue, our approach aims to harmonize the learning process across all classes, ensuring equitable representation and enhancing feature separability.

Contrastive learning[2] complements this by honing on the nuances that distinguish similar from dissimilar data points, a critical factor in mitigating the adverse effects of class imbalance. Our methodology is rooted in a multifaceted strategy where metric learning combined with cross-entropy[1] fosters an equitable learning environment across diverse classes, ensuring no class is overshadowed regardless of its frequency within the dataset. This composite loss function is designed to bolster model robustness and generalization capabilities, addressing the inherent challenges of the LTR problem comprehensively.

The validation of our proposed solution was conducted through rigorous experimentation on modified CIFAR-10/100 [3] datasets and a bespoke custom dataset, showcasing our approach's adaptability and effectiveness across varying levels of class imbalance. Utilizing ResNet models [4] of differing depths (ResNet18, ResNet34, and ResNet50) and experimenting with a range of loss functions—including focal loss[5] and supervised contrastive[2] loss—allowed us to assess our methodology's performance in a broad spectrum of scenarios. This experimental setup not only facilitated a deep dive into the comparative analysis of model behaviors but also enabled the identification of optimal configurations for tackling LTR challenges.

Our findings illustrate significant improvements in model performance, particularly in environments characterized by pronounced class imbalances. By employing our integrated approach, we were able to set new benchmarks for image classification models, demonstrating superior performance in handling long-tailed distributions. The study's implications extend beyond the immediate advancements in LTR; it lays a foundational framework for future research in machine learning, emphasizing the importance of a balanced and nuanced approach to model training and development.

Furthermore, our study elucidates the critical role of selecting appropriate loss functions and data augmentation strategies, tailored to the unique characteristics of each dataset. This insight is instrumental in advancing the field of machine learning, guiding practitioners in the development of more robust, equitable, and effective models. Through a comprehensive exploration of the challenges and solutions associated with LTR, this research contributes to a deeper understanding of class imbalance issues, paving the way for innovative approaches in the domain of imbalanced learning. Please find our project on https://github.com/Bish-Soli/MQP.git

Acknowledgements

I would like to express my sincere gratitude to Professor Zimming Zhang for their invaluable guidance and unwavering support throughout the course of this research project. Their expertise and mentorship have been instrumental in shaping the direction and quality of this work.

I would also like to thank Worcester Polytechnic Institute for providing the resources, environment, and academic foundation that made this project possible. This institution's commitment to excellence in education and research has played a pivotal role in my academic journey.

This project would not have been achievable without the contributions and support of both my professor and my institution.

Contents

List of Tables

List of Figures

1 Introduction

Long-tailed data distributions pose a unique challenge for the field of image classification [7]. This environment, characterized by significant class imbalance [8][9] raises critical questions about the efficacy of standard image classification methods. In these scenarios, head classes (with abundant data) tend to overshadow tail classes (with sparse data); leading to models which often prioritize abundant head classes, neglecting rare tail classes due to data scarcity. The resulting bias is dangerous in safety-critical environments and thus hinders real-world applicability. Artificial intelligence has shown remarkable success in image classification, excelling in identifying and categorizing objects within images [10]. This success is largely due to advanced neural networks and large, well-balanced training datasets [11]. However, AI's performance drops significantly when trained on imbalanced datasets. These systems tend to hold bias toward majority classes, leading to poor recognition of minority classes seen in Figure1 1 [12][8] . This challenge is particularly evident in real-world datasets where class imbalance is common.

1.1 Trustworthy AI

Therefore, despite the recent technological advancements, early adopters of AI in safety-critical environments are cautious - citing concerns about accuracy, bias, and the ability to handle diverse scenarios [13]. The field of trustworthy AI stands to address these concerns, emphasizing fairness through explainability, transparency, and robustness. Trustworthy AI is ripe for fields such as medical imaging [14] or wildlife monitoring, whereby overlooking rare but critical categories could have severely detrimental consequences.

1.1.1 Long-Tailed Recognition

Encompassing robustness and fairness, long-tailed recognition (LTR) is a step in this direction. While there exist strategies to mitigate class imbalance, such as data resampling [15] and loss re-weight [16], they often lead to performance trade-offs. Recent research efforts have shifted towards more sophisticated approaches, focusing on creating a representative feature space and ensuring equitable treatment of all classes. This shift to the field of LTR marks a significant step forward in creating more robust and fair image classification models in long-tailed data environments. The essence of LTR is to develop models that can effectively learn from and accurately classify images from both head and tail classes, ensuring balanced performance across all categories [17][18][19][20]. Success in this field is vital in critical applications to enhance the capability of AI systems in handling diverse and realistic datasets, especially those involving

rare events or minority classes. Therefore, while AI is currently employed in safety-critical environments[21], its adoption at scale varies by industry and application. Demonstrating AI's ability to mitigate the challenges posed by long-tailed distributions in data can accelerate its broader adoption across various critical sectors. Furthermore, progress in this field will spur inclusive and equitable technological solutions.

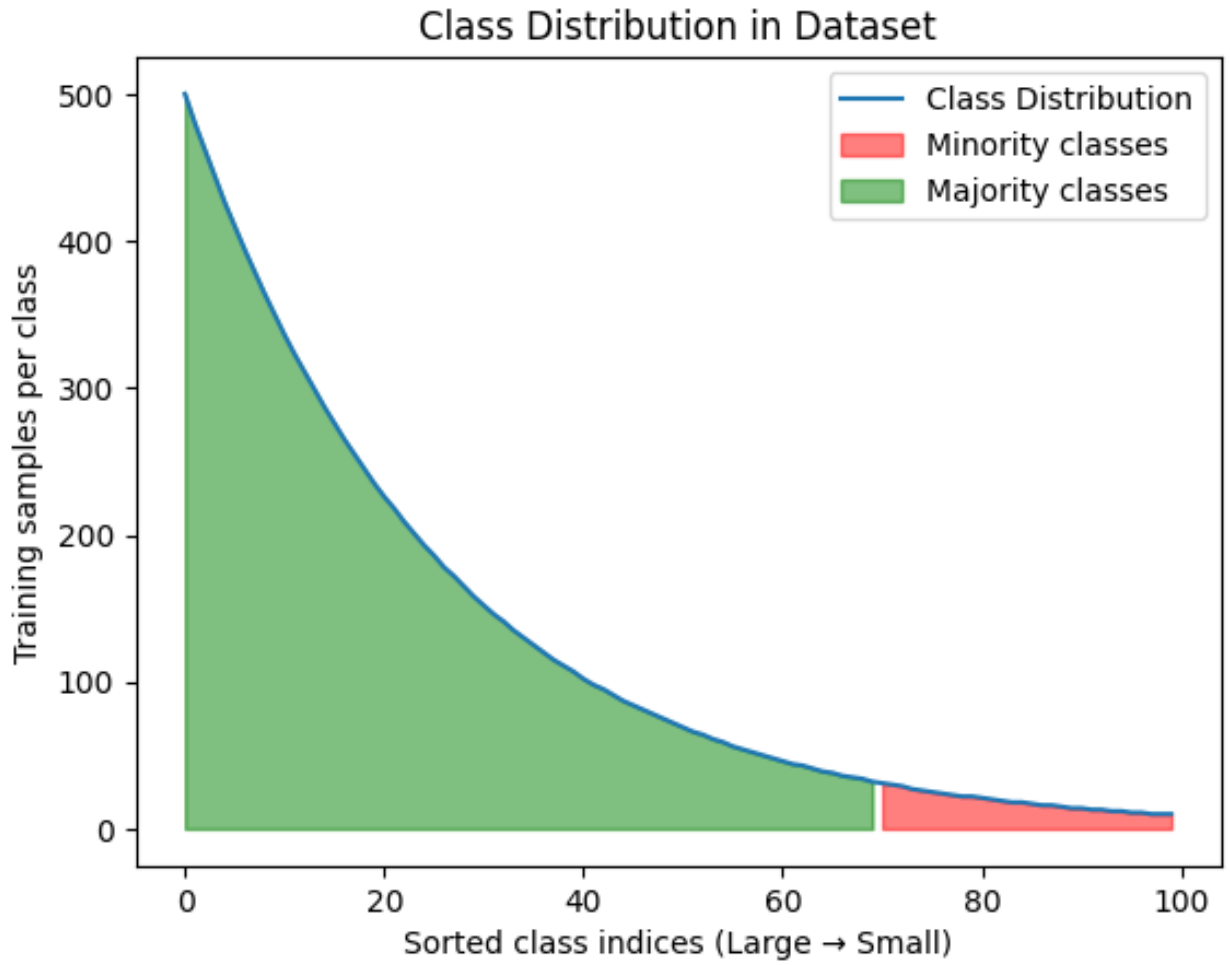

Figure 1: An illustration of the long-tailed distribution in image datasets and the challenge it presents in classification tasks. This figure highlights the disparity between head (majority) classes and tail (minority) classes, emphasizing the importance of developing effective strategies for long-tailed recognition.

1.2 Contributions

In this paper, we commence with a comprehensive review of related work in the domain of long-tailed image recognition, with a special emphasis on latent space and feature-learning methods. This exploration is critical as it underpins the development of more effective and equitable models in this field. Additionally, we delineate the datasets utilized in our research, offering insights into their composition and relevance to our

study. Following this, we delve into a detailed exposition of the methodologies employed in our experiments, particularly highlighting the roles of contrastive learning [22].

The paper culminates with the proposition of variety of solid methods , offering a composition of independently various techniques. The integration of these approaches enhances the model's capability to handle data imbalance inherent in long-tailed distributions. A thorough analysis of the results are presented, highlighting the efficacy of our approach, followed by a discussion on potential future research directions, aiming to further the frontiers of long-tailed image recognition.

2 Related Works

2.1 Image Classification

The use of convolutional neural networks (CNNs)[23] for image classification has been a significant advancement in computer vision [7][10][24]. The ResNet architecture[4], introduced by , has significantly contributed to the field of image recognition through its innovative approach to residual learning, which enables the training of very deep networks by utilizing skip connections to bypass one or more layers[24]. Similarly, the VGG network architecture[25], as demonstrated by and Zisserman, has been influential due to its simplicity and effectiveness, making it well-suited for image classification tasks, particularly on benchmark datasets such as ImageNet [26]. Moreover, the training process for image classification typically involves minimizing a cross-entropy loss[1] function, which measures the disparity between predicted class probabilities and the true distribution of classes in the training data. Additionally, advancements in image classification have been driven by improvements in data sampling techniques and architectural changes, resulting in the extraction of more discriminative features from images and ultimately improving the overall performance of image classification models [27].

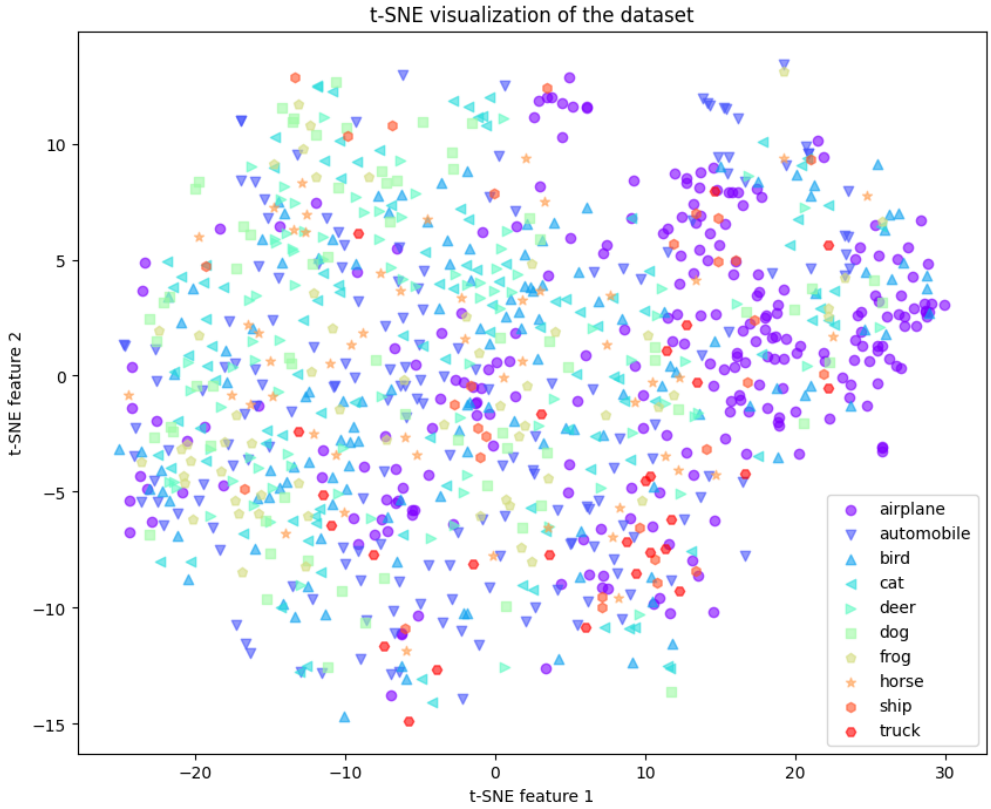

Figure 2: t-SNE visualization[6] of the CIFAR10 dataset, demonstrating the clustered regions of features corresponding to different classes. Each color and marker type represents a different class, illustrating the separability and overlap between classes in the feature space. This visualization aids in understanding the complexity of the classification problem and the importance of discriminative feature learning.

2.2 Data Augmentation

Data augmentation is a technique used to artificially expand a dataset by creating modified versions of the existing data instances. This process involves applying transformations such as rotation, flipping, scaling, and adding noise to the original data. Data augmentation is employed to address the issue of limited training data, particularly in scenarios where the available dataset is insufficient to effectively train a deep learning model. By generating diverse variations of the original data, data augmentation aims to enhance the generalization capability of the model and reduce the risk of over-fitting, ultimately leading to improved performance on unseen data [28][29]. In the context of imbalanced datasets, data augmentation serves as a typical approach to mitigate the class imbalance problem. However, it is important to note that while data augmentation can alleviate the imbalance to some extent, it does not provide a comprehensive solution to the problem because it primarily generates more data from existing instances without fundamentally changing the class distribution or introducing genuinely new and diverse examples for underrepresented classes. It helps to some extent by increasing the quantity of data for minority classes through transformations, but it does not address the root causes of imbalance, such as the lack of variety and representativeness of minority classes in the dataset.

As a result, researchers have sought out alternative methods to address the challenges posed by imbalanced datasets, such as the utilization of advanced sampling techniques and the development of specialized algorithms tailored to handle class imbalances [30][31]. In the context of our research, data augmentation plays a crucial role in addressing the limitations posed by imbalanced datasets in image classification tasks. By artificially expanding the dataset through diverse transformations, we aim to enhance the robustness of our classification model and improve its performance, particularly in scenarios where certain classes are underrepresented. Furthermore, our study integrates advanced data augmentation techniques to effectively handle the challenges associated with imbalanced datasets, thereby contributing to the development of more robust and accurate image classification models [32][33].

2.3 Data Sampling

In the endeavor to tackle the prevalent issue of class imbalance in long-tailed recognition, Liu et al. leverage an innovative sampling technique [34] to increase robustness for LTR models. The adversarial, class-balanced sampling ensures equitable representation of all classes, including those with fewer examples. By generating synthetic samples for the minority class, the approach yields a balanced training dataset, thereby enhancing the model's performance and robustness against adversarial perturbations [35][36]. This

plays a crucial role in addressing the challenges posed by LTR. This ultimately improves the model's ability to generalize which is crucial for widespread adoption.

2.4 Robust Feature Learning

Features are distinctive attributes extracted from data that enable the identification of objects. Convolutional neural networks (CNNs)[23] utilize features to accurately classify images by learning hierarchical representations of visual data. CNNs[23] employ convolutional filters to extract features from input data, enabling the network to capture intricate patterns and structures. However, conventional features cannot characterize a class with inherent variability. For this reason, robust feature learning aims to identify features that are consistent and reliable within each class, regardless of the intraclass diversity. This is achieved by focusing on learning features that remain unchanged (invariant) under different conditions and variations, which helps the model to perform well not only on the majority classes with abundant data but also on the minority classes with fewer examples[37][38]. To further enhance our understanding of this field, it's crucial to recognize other prevalent methods that contribute to addressing these challenges.

2.5 Latent Space Design

Moreover, latent space plays a crucial role in machine learning, especially within neural networks, serving as the foundational bedrock for understanding and manipulating data representations. Latent space refers to an abstract, multi-dimensional representation of complex data, where relationships and features are encoded in a way that is not immediately apparent in the raw data [39] see Figure 22. This space is often where the most critical and intricate aspects of data are captured, enabling neural networks to perform tasks like classification, regression, and even generative processes more effectively. The embedding space is a result of the compressed representation of the input image and the loss function. Specifically, the encoder portion of a neural network is responsible for obtaining the best features that minimize the loss function. In the context of deep learning, an encoder is a set of layers that processes the input data. The goal of these layers is to compress the data into a lower-dimensional space, the latent space, where the most salient and useful features of the data are retained [40]. This process is essential for tasks that require a high level of abstraction, such as image and speech recognition, natural language processing, and more. The design of a robust latent space is crucial for the success of downstream tasks. A well-crafted latent space allows for extraction of robust embeddings, which in turn enhances the performance of the neural network in various applications [41]. Various AI methods have been developed to create effective latent spaces. One of the key

methods is contrastive learning[2], a technique that learns presentations by comparing and contrasting pairs of data points. By doing so, it helps in creating a more defined and structured latent space.

2.5.1 Contrastive Learning

Contrastive Learning[2] is a method in machine learning that focuses on understanding how similar or dissimilar data points are from each other. Fundamentally, it operates on the principle of learning representations by contrasting positive pairs against negative pairs. In simpler terms, it brings representations of similar items closer together and pushes apart those of dissimilar items in the latent space. This technique is particularly effective in scenarios with complex data sets, such as image and speech recognition, where it can discern subtle differences and similarities. Its intuitive approach of handling data, especially in unsupervised or self-supervised learning scenarios, makes it a powerful tool for feature extraction and representation learning [42][43][44][45].

The effectiveness and nuances of Contrastive Learning[2] are well-articulated by Li et al. they tackle the challenge of poor class separability in feature space due to the class imbalance inherent in long-tailed distributions [22]. Their solution, Targeted Supervised Contrastive Learning[2] (TSC)[22], aims to improve the uniformity of feature distribution on a hypersphere by assigning uniformly distributed targets to each class during training. TSC first computes the optimal positions for targets in the feature space, which are distributed on a unit hypersphere (meaning each feature vector has a norm of one). These targets are precomputed before training and remain fixed. The unit hypersphere for target generation in feature space helps normalize feature vectors, maintaining a consistent scale across all classes. This uniformity in feature representation is crucial for mitigating class imbalance in long-tailed distributions.

The predecessor to TSC[22], KCL[46] , is a machine learning approach focused on improving the uniformity of feature distribution across different classes in long-tailed recognition tasks. It achieves this by uniformly distributing class centers on a hypersphere, enhancing class separability and decision boundaries. . For the CIFAR-10-LT dataset, TSC[22] achieves an accuracy of 76.5%, which is a substantial improvement over the KCL method[46], which achieves an accuracy of 71.8%.

Zhu et al. introduce Balanced Contrastive Learning (BCL)[47], which enhances the performance of both head and tail classes [47]. BCL[47] works by combining two primary components: class-complement and class-averaging techniques. These techniques are integrated with the supervised contrastive learning framework[2]. The class-complement technique focuses on enhancing the feature learning for tail classes by using the negative samples from head classes. In contrast, class-averaging balances the contribution of each

class to the loss function, thereby mitigating the dominance of head classes. The results of their approach are impressive. BCL[47] achieves a top-1 accuracy of 64.3% for many, 37.1% for medium, and 8.2% for few-shot learning, with an overall accuracy of 43.7% on long-tailed CIFAR [3] datasets.

2.5.2 Geometric Design

Also in machine learning, the geometric design of a latent space, such as spherical or Euclidean geometry, plays a crucial role in how data is represented and processed. Spherical geometry provides several benefits to generalization, as it encourages a structured and well-distributed latent space. The symmetry gained by having data points distributed on a sphere's surface, making each point equidistant from the center, reduces the likelihood of overfitting to specific patterns in the training data. This design is advantageous in contrastive learning[2], as it offers a meaningful measure of data point similarity, crucial for applications like image and speech recognition. Conversely, Euclidean geometry positions data in a traditional Cartesian space, favoring tasks that benefit from linear relationships and straightforward distance calculations, like clustering and linear separability in neural networks. The choice between these geometries hinges on the nature of the data and the specific requirements of the task, each offering distinct benefits in terms of data representation and model performance[48][49].

Yanbiao et al. propose a novel approach involving curvature regularization to balance the perceptual manifolds in deep neural networks [50]. The authors introduce a curvature regularization term into the loss function that modifies the latent space geometry of deep neural networks. This approach aims to flatten the feature manifolds, reducing the bias towards head classes in long-tailed distributions. By adjusting the curvature of the latent space, they ensure a more balanced representation for both head and tail classes.

Liu et al. introduced a methodology centers on Geometric Structure Transfer (GIST) [51]. This involves encoding the geometric structure of well-represented (head) class features into a constellation of classifier parameters, which are then transferred to aid in recognizing under-represented (tail) classes. The network learns to map these geometric structures across different classes, enabling it to leverage the rich feature information from head classes to improve the recognition of tail classes.

2.5.3 Other

Cui et al. propose a novel approach to address the long-tailed recognition problem in visual datasets [16]. Their method, ResLT, uses a residual learning mechanism to rebalance the parameter space between head (frequent) and tail (rare) classes. The ResLT framework includes parameter specialization and a

residual fusion module. The former allocates individual parameters specifically for tail classes. This allows the model to effectively learn and represent features from these less frequent classes. Furthermore, the multi-branch structure enables nested class assignments and different focuses for each branch. These branches help compensate for the under-representation of tail classes.

3 Datasets

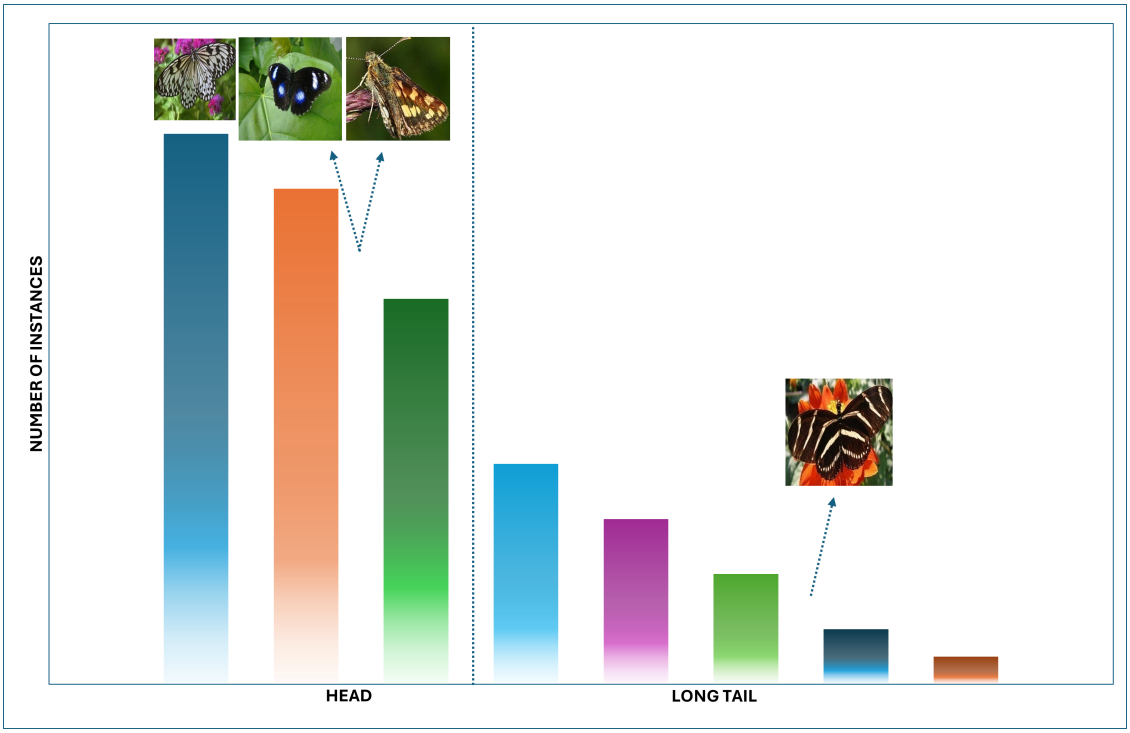

Figure 3: Imbalanced Dataset distribution

3.1 CIFAR-10/100 LT

In our study, we leverage the CIFAR-10 [3] and CIFAR-100[3] datasets, each comprising 60,000 images segmented into 50,000 images for training and 10,000 images for validation. CIFAR-10 categorizes these images into 10 distinct classes, facilitating a broad spectrum of basic object recognition tasks. On the other hand, CIFAR-100[3], with its division into 100 classes, introduces a more granular classification challenge, encompassing a wider variety of objects and thereby increasing the complexity of the recognition task.

For the purpose of our experiment, we introduce a strategic modification to these datasets to sim-

ulate long-tailed distributions—a scenario that closely mimics real-world data challenges where some classes are significantly underrepresented compared to others. This transformation is crucial for our investigation as it allows us to explore the efficacy of our proposed approach in a context that is notoriously difficult for conventional image classification models due to the inherent imbalance. The creation of these long-tailed versions of the CIFAR[3] datasets is meticulously designed to reflect a realistic skew in class distribution. we follow Zhu et al. code [47] https://github.com/FlamieZhu/Balanced-Contrastive-Learning in creating the Imbalanced dataset. Utilizing a custom script, we systematically reduce the sample size for each subsequent class, maintaining a decreasing representation from the most populous (head) classes to the least populous (tail) classes. This deliberate skewing is achieved through a calculated modulation of the number of samples per class, ensuring a progressive diminution that accurately embodies long-tailed phenomena. Reference Figure 3.

This methodical adjustment not only challenges the model's ability to learn from limited data but also tests its robustness and generalization across diverse class distributions. By engaging with this skewed dataset, our research confronts the critical issue of class imbalance head-on, addressing a pivotal hurdle in the advancement of equitable machine learning models capable of recognizing and classifying a wide array of images, regardless of their frequency in the training dataset.

3.2 Creating Imbalaced Dataset

We follow Zhu et al. method of creating the Imbalanced Dataset[47]. The **dataset.py** script is a cornerstone of our project, designed to facilitate the study of class imbalance effects within image classification tasks. It introduces a framework for generating imbalanced versions of the CIFAR10[3] and CIFAR100[3] datasets, alongside capabilities to manage custom datasets defined by a directory structure. This script encapsulates both the architecture and operations necessary to manipulate dataset distributions and evaluate model performance under varying degrees of class imbalance.

3.2.1 Core Classes and Architecture

At the heart of this framework is the **ImbalanceDataset** class, a versatile base class crafted to generate imbalanced datasets. It is adept at handling not only the **CIFAR10**[3] and **CIFAR100**[3] datasets but also custom datasets supplied through a directory path. Initialization parameters for this class include the **CIFAR**[3] version, the dataset's root directory, indicators for training or testing subsets, data transformations to be applied, the imbalance ratio to achieve, and an optional path for custom datasets.

This design enables a flexible and dynamic approach to dataset preparation, allowing for precise control over the degree of imbalance introduced.

3.2.2 Key Functionalities

The script introduces a method, **_create_long_tailed**, dedicated to adjusting the class distribution to form a long-tailed curve, emblematic of imbalanced datasets. By computing and then modulating the number of samples per class, it ensures a systematic reduction in class representation from the most to the least populous, mimicking real-world data scenarios where some classes are naturally less frequent.

Dataset initialization is a critical function, determined by the provided parameters, which dictate whether the **CIFAR**[3] datasets are loaded via torchvision or from a custom directory. In the case of custom datasets, the script adeptly enumerates the classes based on the directory structure. A notable feature is the inclusion of a debug mode, which constrains the dataset size for expedited experimentation and debugging processes. Data transformation capabilities are embedded within the class, supporting a sequence of operations like normalization and augmentation to be applied during dataset loading, thereby ensuring the data is suitably prepared for model training and evaluation. Expanding on the **ImbalanceDataset** base class, the script also defines **ImbalanceCIFAR10** and **ImbalanceCIFAR100** subclasses. These are specifically optimized to generate imbalanced versions of their respective **CIFAR**[3] datasets, leveraging the foundational functionalities of the base class to introduce the desired degree of imbalance.

3.2.3 Additional Utility Functions

To complement dataset creation and manipulation, the script includes a suite of utility functions aimed at data visualization and analysis. Functions such as **imshow**, **show_augmented_images**, **plot_class_distribution**, and **plot_class_distribution_imbalanced** offer insights into the dataset's structure and the impact of imbalance adjustments. Moreover, for in-depth feature analysis, functions like **extract_features_and_labels**, **visualize_with_tsne, and visualize_with_pca** are provided, facilitating the exploration of data separability and the high-dimensional landscape of features through techniques like **t-SNE** and **PCA**.

Dataset sampling methods, including **get_random_batch** and **get_samples_from_each_class**, are integral to the script, enabling the selection of balanced or random data samples for model training, testing, or inspection. This functionality is crucial for assessing model performance across varied class distributions and ensuring balanced exposure to all classes during the training process.

3.2.4 Usage

Utilizing the classes and functions within **dataset.py** typically involves instantiating an **ImbalanceCIFAR10** or **ImbalanceCIFAR100** object with specific parameters, such as the imbalance ratio and desired transformations. This object can then seamlessly integrate with a PyTorch DataLoader, creating an iterable data loader that facilitates the training or evaluation of machine learning models under conditions of class imbalance.

3.3 Butterfly Custom Dataset

The Butterfly Custom Dataset used in this project was sourced from Kaggle. It can be found at the following URL: Kaggle Butterfly Image Classification Dataset. This dataset comprises a total of 6500 samples, distributed across 75 classes. The distribution of the samples is as follows: 80:20 for training and testing. To prepare the Butterfly Custom Dataset, the initial dataset was first unzipped within a Google Drive environment. Subsequently, a systematic methodology was applied to organize the dataset into an imbalanced structure suitable for our study. The procedure involved reading the dataset's metadata from a CSV file, which provided essential details such as file names and associated labels. Based on this metadata, the dataset was divided into two subsets: a test set and a training set. For each class label, we created separate directories within both test and training folders to maintain the dataset's structure. Eight images were allocated to each class in the test set, with the remainder being used for the training set. In our case, the top eight images from each class were selected for the test set to simulate a scenario where the dataset would exhibit a natural imbalance. This selection process was carried out for each class, effectively creating an imbalanced dataset that mirrors real-world conditions where certain classes are underrepresented. The images were then programmatically moved to their respective directories based on the organization defined by the metadata. Reference Figure 4.

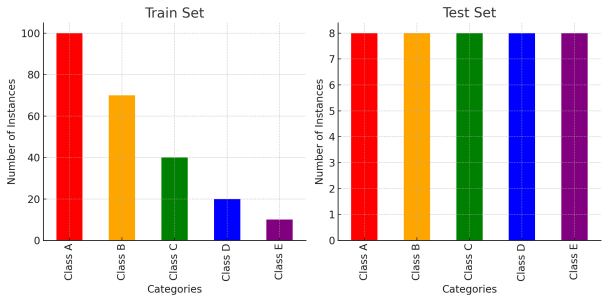

Figure 4: Butterfly Custom Dataset distribution

3.3.1 Reason for Dataset Selection and Image Size

The Butterfly Custom Dataset was specifically chosen to evaluate the effectiveness of our methodology on real-world image datasets. This decision was made to move beyond the constraints of commonly used, predefined datasets such as CIFAR-10/100[3], which, while standard, do not always represent the complexity and variability found in real-world scenarios. By selecting this particular dataset, we aim to demonstrate the applicability and robustness of our approach in a more varied and realistic context.

Furthermore, each image in the dataset has been standardized to a resolution of 224 x 224 pixels. This uniform size was selected to conform with the input size requirements of prevalent deep learning architectures while also ensuring that the images retain sufficient detail necessary for accurate classification. It strikes a balance between computational efficiency and the preservation of visual information critical for the performance of the models being evaluated. As a preprocessing step, each image was resized to a 64x64 using bilinear interpolation.

Algorithm 1 Algorithm for Creating Imbalanced Datasets[47]

procedure CREATEIMBALANCEDDATASET($datasetType, rootDir, train, transform, imbRatio, datasetPth$)
    Initialization:
    if $datasetType =$ "CIFAR10" or $datasetType =$ "CIFAR100" **then**
        Download and load the CIFAR dataset.
    else if $datasetPth \neq$ NULL **then**
        Load custom dataset from $datasetPth$.
    end if
    Determine number of classes $numClasses$.
    Initialize empty lists for $indices$, $targets$.
    Creating Long-Tailed Distribution:
    Calculate class distribution $classCounts$.
    for $class = 1$ **to** $numClasses$ **do**
        $numSamples = classCounts[class] \times imbRatio^{\frac{class-1}{numClasses-1}}$
        Select $numSamples$ indices for $class$ randomly.
        Append selected indices and class labels to $indices$, $targets$.
    end for
    Shuffle $indices$ to mix class samples.
    Utility Functions:
    Implement $\_\_len\_\_()$, $\_\_getitem\_\_(idx)$, and other utility functions as required.
    Return dataset object with methods overridden for imbalanced data retrieval.
end procedure

4 Methods

4.1 Problem Definition

The primary focus of our investigation is the challenge of class imbalance in unbalanced datasets. such imbalance often skews a model's learning process, leading to over-fit to the majority classes while underrepresenting the minority ones. The goal is to develop and refine methodologies that effectively address this imbalance.

4.2 Weight Balancing

Weight balancing in Long-Tailed Recognition (LTR) is crucial for addressing class imbalance in machine learning datasets. This technique involves adjusting the training process by assigning different weights to classes in the loss function, giving more weight to underrepresented classes and less to overrreprented ones. This compensates for skewed class distribution, reducing bias towards the majority and improving model performance across all classes[52]. Long-tailed recognition methods can be categorized into class-imbalanced learning, logit adjustment, and ensemble learning[53].The imbalance in classification data causes poor per-

formance in the minority class, and various techniques such as data re-sampling and loss re-weighting are used to address this issue [54]. The drawback to this approach is it can lead to overfitting to minority classes and may not effectively address the root cause of imbalance. Determining optimal weights for classes is challenging and can result in sub-optimal performance, especially in extremely skewed distributions. This approach, while helpful, often requires additional strategies to ensure balanced learning and generalization.

4.2.1 Weight Decay

Weight decay serves as a regularization strategy in Long-Tailed Recognition (LTR), where it helps to counteract the overfitting often seen with imbalanced class distribution. By adding a penalty term to the loss function, proportional to the sum of the squared weights of the model,weight decay encourages simpler, more generalized models. The key intuition is to promote smaller, generalized weights, reducing the model's tendency to learn complex patterns specific to frequent classes, a common challenge in LTR [52][11]. The drawback to this approach is it can sometimes be too restrictive, potentially leading to under-fitting if the penalty on the model's complexity is too high. Additionally, finding the optimal level of weight decay is challenging and requires careful tuning to avoid adversely impacting model performance.

4.2.2 Max Normalization

Max Normalization is another technique used to scale input features or model weight. max normalization involves scaling the features or weights by dividing them by the maximum value in the dataset or layer. this process ensures that all inputs or weights have a constant scale, mitigating the risk of certain features disproportionately influencing the model due to their large magnitude. The intuition is to normalize the influence of each feature or weight , promoting a more balanced learning process where no single class dominates due to scale difference[55][52]. While effective in scaling features or weights uniformly, can sometimes oversimplify data representations, potentially losing critical information in the process. This technique may not adequately address inherent data complexities or class imbalances in datasets. Furthermore, it assumes that the maximum value is an appropriate scale for all features, which might not always align with the specific nuances of the data.

4.3 Metrics Learning

Metric learning is pivotal in LTR. This approach focuses on learning a distance function that effectively measures the similarity or dissimilarity between data points. in the context of LTR, where minority

classes are underrepresented, metric Learning aims to learn a space where distances between similar items (regardless of their class frequency) are minimized, and those between dissimilar items are maximized. the intuition is to create a feature space that enhances the separability of all classes, making it easier for the model to distinguish between them, particularly for those classes with fewer examples. by doing so it insures that the model is equally sensitive to all classes, enhancing the overall accuracy and fairness[56].

Though this approach is powerful in enhancing class separability, can be complex to implement and may require substantial computational resources. It also faces the challenge of defining an appropriate distance metric that accurately captures the nuances of different classes, especially in highly diverse datasets.

Given an input x, the feature representation is obtained by passing it through an encoder:

$$f = \text{Encoder}(x)$$

These features are then linearly transformed to raw class scores (logits):

$$z = Wf + b$$

where W is the weight matrix and b is the bias vector of the fully connected layer. The predicted probabilities for each class are obtained using the softmax function:

$$p_{o,c} = \frac{e^{z_c}}{\sum_{j=1}^{M} e^{z_j}}$$

Finally, the cross-entropy loss[1] for a single observation is calculated as:

$$L = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$$

where $y_{o,c}$ is a binary indicator of whether class c is the correct classification for observation o, and M is the number of classes.

4.4 Contrastive Learning

Contrastive Learning[2] is an advanced powerful method effictive in LTR, where managing class imbalance is crucial. This technique emphasizes learning representations by contrasting positive (similar)

pairs against negative (dissimilar) pairs. In LTR scenarios, Contrastive Learning[2] aims to ensure that samples within the same class (intraclass) are brought closer together in the feature space, while samples from different classes (interclass) are pushed apart. This method hinges on the idea that a model can learn more robust and discriminative features by focusing on the similarities within classes and the differences between them, regardless of class frequency[57][22].

While this approach is effective in distinguishing between similar and dissimilar data points, can require large amounts of data and computational resources for training. Additionally, it may struggle with ambiguous or overlapping class boundaries where the distinction between classes is not clear-cut.

The supervised contrastive loss[2] is used for this purpose and can be defined mathematically as:

$$L_{contrastive} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(\text{sim}(z_i, z_p)/\tau)}{\sum_{a \in A(i)} \exp(\text{sim}(z_i, z_a)/\tau)} \tag{1}$$

where I is the set of all images, $P(i)$ is the set of positives for image i, z represents the normalized embeddings, sim is the cosine similarity, and τ is the temperature parameter.

4.5 Focal loss

The Focal Loss[5] is an enhanced version of the Cross-Entropy Loss[1] designed to address class imbalance by assigning more weight to hard, or easily misclassified examples and less weight to easy examples. It introduces a modulating factor to the standard cross-entropy[1] criterion that forces the model to focus on hard examples. This adds a factor $(1 - \hat{y}_t)^\gamma$ to the standard Cross-Entropy Loss[1], which reduces the relative loss for well-classified examples, putting more focus on hard, misclassified examples. Additionally, a weighting factor α_t can be applied to deal with class imbalance. The complete formula is given by:

$$L_{focal}(y, \hat{y}) = -\alpha_t (1 - \hat{y}_t)^\gamma \log(\hat{y}_t) \tag{2}$$

where $\hat{y}_t$ is the predicted probability of the true class t, γ is the focusing parameter, and α_t is a scalar or vector of weighting factors for each class.

5 Approach

5.1 Experimental Setup

This section meticulously outlines our experimental strategy designed to evaluate the impact of different loss functions on model performance across various imbalanced datasets. We employ a comprehensive suite of experiments leveraging three ResNet architectures[4] (ResNet18, ResNet32, ResNet50) across three distinct datasets (CIFAR-10[3], CIFAR-100[3], Butterfly Customer Dataset). For each model-dataset combination, we explore the following loss functions:

- Cross-Entropy Loss[1]
- Focal Loss[5]
- Supervised Contrastive Loss with $\alpha = 0.5$, $\beta = 0.5$

5.1.1 Computational Environment

All experiments are conducted using Google Colab, equipped with NVIDIA A100 GPUs, ensuring high computational efficiency and enabling the handling of large-scale datasets and extensive model architectures.

5.1.2 ResNet Models

- **ResNet18**[4]: This model is built with a series of convolutional neural networks[23] and uses the BasicBlock architecture. It's a smaller version of ResNet[4], making it faster and less computationally intensive, suitable for less complex datasets or when computational resources are limited.
- **ResNet34**[4]: Similar to ResNet18[4], this model also utilizes BasicBlock but with a deeper architecture. It strikes a balance between computational efficiency and model complexity.
- **ResNet50**[4]: Utilizing the Bottleneck architecture, ResNet50[4] is significantly deeper, allowing it to learn more complex features. It's designed for more complex datasets and tasks but requires more computational power.

In our experimental setup, we aim to assess the performance of various models on different datasets. Specifically, we are interested in understanding how model depth, such as that seen in the variations be-

tween ResNet18[4], ResNet34[4], and ResNet50[4], impacts the feature extraction capabilities and the overall classification accuracy.

ResNet18[4], ResNet34[4], and ResNet5[4]0 represent three points in the spectrum of depth for convolutional neural networks[23]. With ResNet18[4] as the shallowest and ResNet50[4] as the deepest, ResNet34[4] offers a middle ground with an increased capacity for feature extraction over ResNet18[4], but with less complexity than ResNet50[4]. The primary expectation is that ResNet50[4], with its increased depth and complexity, would be able to learn the most nuanced features from the data. However, ResNet34[4] might provide a balance between depth and computational efficiency, potentially offering the best trade-off for certain types of image datasets. This tiered approach to model depth is based on the hypothesis that deeper models have a greater capacity for representing complex functions and hierarchies of features. This is particularly beneficial when dealing with high-resolution images or intricate patterns within the data. Each model's performance will be critically evaluated to determine the optimal balance between depth and efficiency, providing insights into how these factors influence the success of image classification tasks in various real-world scenarios.

Regarding the loss functions—Focal loss[5], Supervised Contrastive loss (SupCon)[2] , Cross-Entropy (CE)[1], and the combination of SupCon and CE, we anticipate varying levels of performance across different scenarios. Focal loss[5] is designed to address class imbalance by focusing more on hard-to-classify examples, making it potentially advantageous in our imbalanced dataset scenario. SupCon, which emphasizes the learning of feature representations by bringing closer samples of the same class and pushing apart samples from different classes in the feature space, might outshine others when the task requires a clear separation of class clusters. The combination of SupCon and CE is expected to leverage the benefits of both contrastive learning[2] and traditional classification objectives. This hybrid approach could potentially yield the best results by ensuring that the model is not only good at distinguishing between different classes but also at making confident predictions for individual instances. In scenarios where inter-class variance is subtle, SupCon might prove superior due to its emphasis on relative comparison between samples. On the other hand, in situations where the classes are well-separated and the challenge lies more in dealing with intraclass variance, focal loss[5] or CE might be more effective. Each approach has its strengths, and the best-performing loss function may vary depending on the specific characteristics of the dataset and the task at hand. For instance, in highly imbalanced datasets, focal loss[5] might outperform others, while in tasks requiring fine-grained distinction between classes, SupCon could provide an edge. Our experiments are designed to shed light on these nuances and offer insights into the scenarios where one method would outperform the other.

These models and loss functions were selected for their representativeness of common approaches in the field and their potential synergistic effects when combined. By exploring these combinations, we aim to derive insights into their applicability and efficacy across different datasets and imbalanced scenarios, thereby informing future research directions and applications in the domain of imbalanced learning.

5.1.3 Constants Across Experiments

To maintain consistency and ensure comparability across experiments, we adhere to the following constants:

- **Epochs:** Each model is trained for 100 epochs, depending on the convergence behavior observed during preliminary experiments.
- **Batch Size:** The default batch size is set to 32 for all experiments to balance computational demand and training stability.
- **Early Stopping:** Implemented with a patience of 10 epochs, this mechanism ceases training if the validation loss does not show improvement, mitigating overfitting risks.
- **Learning Rate:** Default set to 0.1. This controls the step size at each iteration while moving toward a minimum of a loss function.
- **LR Decay Epochs:** Specifies the epochs at which the learning rate should decay. Default values are '700,800,900'.
- **LR Decay Rate:** The factor by which the learning rate should decay. Default is 0.1.
- **Momentum :** Default set to 0.9, it accelerates the gradient vectors in the right direction, leading to faster converging.
- **Temperature:** Controls the sharpness of the softmax distribution in contrastive learning[2], set to 0.07 by default.

5.1.4 Loss Functions and Mathematical Formulas

1. **Cross-Entropy Loss**: The Cross-Entropy Loss[1] is fundamental for classification tasks, aiming to minimize the distance between the true distribution y and the predicted distribution $\hat{y}$. The formula

for Cross-Entropy Loss[1] is given by:

$$L_{CE} = -\sum_{c=1}^{M} y_{o,c} \log(\hat{y}_{o,c}) \tag{3}$$

where M is the number of classes, $y_{o,c}$ is a binary indicator of whether class c is the correct classification for observation o, and $\hat{y}_{o,c}$ is the predicted probability of observation o being of class c.

2. **Focal Loss**: Focal Loss[5] addresses class imbalance by focusing more on hard-to-classify examples. It is defined as:

$$L_{FL} = -\alpha_t (1 - \hat{y}_t)^{\gamma} \log(\hat{y}_t) \tag{4}$$

where α_t is the weighting factor for class t, γ is the focusing parameter that adjusts the rate at which easy examples are down-weighted, and $\hat{y}_t$ is the predicted probability for the true class t.

3. **Supervised Contrastive Loss**: This loss enhances feature learning by promoting class separability in the embedding space. For $\alpha = 1.0, \beta = 0.0$, it is calculated as:

$$L_{SCL} = \frac{1}{N} \sum_{i=1}^{N} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(\text{sim}(z_i, z_p)/\tau)}{\sum_{a \in A(i)} \exp(\text{sim}(z_i, z_a)/\tau)} \tag{5}$$

For a combination when $\alpha = 0.5, \beta = 0.5$, the loss is adjusted to incorporate both components weighted by α and β. In these formulas, N is the number of samples, $P(i)$ denotes the set of indices of positive samples for i, $A(i)$ includes all other indices, sim denotes the cosine similarity, z represents embeddings, and τ is a temperature scaling parameter.

5.1.5 Hypotheses and Model Selection Rationale

To systematically investigate the performance of various models on imbalanced datasets, we have formulated several hypotheses. Given the depth variations between ResNet18[4], ResNet34[4], and ResNet50[4], we hypothesize that the deeper architectures will exhibit superior performance in extracting nuanced features from the datasets, potentially leading to better classification accuracy. However, this may come at the cost of increased computational resources and overfitting risks, especially for ResNet50[4].

We expect ResNet18[4] to perform adequately on less complex datasets due to its smaller architecture, while ResNet34[4] might strike a balance between performance and efficiency. For ResNet50[4], which is significantly deeper, we anticipate it to excel on more complex datasets where the abstraction of higher-level features is critical.

As for the loss functions—cross-entropy (CE)[1], focal loss[5], supervised contrastive loss (SupCon)[2] , and the combination of SupCon + CE—we hypothesize that focal loss[5] might perform better on datasets with severe class imbalance by emphasizing the learning of underrepresented classes. SupCon is expected to enhance the separability of classes in the feature space, which might be more beneficial for datasets where classes are not well-separated. The hybrid approach of SupCon + CE could potentially combine the strengths of both, offering improved performance across various scenarios.

5.1.6 Experimental Validation

Upon completing our experiments, we will compare the actual outcomes against our hypotheses. This will not only validate our initial predictions but also shed light on the dynamics between model complexity, loss function selection, and dataset characteristics. The findings will contribute to a more nuanced understanding of model behavior in the context of imbalanced image classification and guide the selection of models and loss functions for specific types of datasets in future work.

5.1.7 Conclusion

This experimental framework is designed to provide deep insights into how different loss functions influence model training and performance on imbalanced datasets. By evaluating these models across a spectrum of datasets with varying levels of class imbalance and architectural complexities, we aim to uncover robust, generalizable insights that can guide future research and applications in imbalanced learning scenarios.

6 Results

6.1 Explanation

In the development and implementation of our model, we adhere closely to the established and groundbreaking architecture known as ResNet (Residual Networks)[4], as introduced by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun in their pivotal work, "Deep Residual Learning for Image Recognition" (arXiv:1512.03385). This architecture has set a new standard for deep learning models, particularly in the domain of image recognition, by enabling the training of significantly deeper networks through the innovative use of residual blocks. These blocks facilitate the flow of gradients during training, thereby alle-

viating the vanishing gradient problem and resulting in enhanced learning capabilities. Our implementation is adapted from the codebase provided at https://github.com/bearpaw/pytorch-classification, which serves as a robust foundation for building ImageNet-style ResNet models[4] in PyTorch.

Furthermore, recognizing the challenges posed by class imbalance in image datasets—a scenario where certain classes are underrepresented compared to others—we incorporate principles from "Targeted Supervised Contrastive Learning[2] for Long-Tailed Recognition" by Li et al., presented at CVPR 2022. This approach reimagines the contrastive learning[2] paradigm, traditionally used in self-supervised settings, for supervised learning to specifically address the long-tail distribution of classes. By leveraging the discriminative power of contrastive learning[2], we aim to enhance the representation of minority classes, thus fostering a more balanced and equitable learning process across all classes. Our methodology combines the robust backbone of ResNet[4] with the targeted application of Supervised Contrastive Learning[2] (SupCon) to navigate the complexities of long-tailed class distributions effectively. This hybrid strategy not only benefits from the deep residual learning's representational depth and efficiency but also from the nuanced approach to addressing class imbalance offered by supervised contrastive learning[2]. The adaptation of these advanced techniques underscores our commitment to pushing the boundaries of accuracy and fairness in image classification tasks.

In summary, our work is grounded in the rich legacy of ResNet's architectural[4] innovations and enriched by the latest advancements in tackling class imbalance through Supervised Contrastive Learning[2]. This dual approach ensures that our models are not only capable of learning deep, complex representations but are also attuned to the nuances of real-world data distributions, making them highly effective for a wide range of image recognition applications.

Dataset Handling

Custom Dataset Preparation **ImbalanceCIFAR Class**: This class handles the creation of imbalanced versions of the CIFAR datasets or custom datasets by specifying a path. It introduces an imbalance by adjusting the distribution of classes based on an **imbalance ratio**, creating a long-tailed distribution to simulate real-world scenarios where some classes are underrepresented.

Data Augmentation

While specific transformations are not detailed in the snippets provided, the dataset class supports the passing of a **transform parameter**, allowing for data augmentation techniques such as random cropping, flipping, and normalization to improve model robustness and generalization.

Model Architecture

ResNet[4] Customization **BasicBlock and Bottleneck**: Two types of blocks are implemented, allowing for the construction of various ResNet[4] configurations. Each block type supports an **is last** parameter, potentially used for models where the final block's output is treated differently, such as in feature extraction scenarios.

Adaptive Components: The ResNet[4] class is designed to be adaptable, with support for custom input channels and optional zero initialization of residual connections, catering to a wide range of image sizes and dataset specifics.

Training Methodology

Contrastive[2] and Supervised Learning

Hybrid Training Approach: The train_contrastive method[2] encapsulates a training process that combines SupCon (Supervised Contrastive Learning[2]) or SimCLR (Simple Framework for Contrastive Learning[2] of Visual Representations) with cross-entropy[1] loss, aiming to leverage the strengths of both contrastive learning[2] for feature space optimization and supervised learning for direct classification.

Learning Rate Warmup and Adjustment: The training scripts incorporate a learning rate warmup strategy, gradually increasing the learning rate from a lower value to facilitate more stable and effective model training in the initial epochs.

Loss Functions

Addressing Class Imbalance **Focal Loss[5] Implementation**: To specifically address the challenge of class imbalance, Focal Loss[5] is employed. This loss function adjusts the cross-entropy[1] loss such that harder-to-classify examples are given more focus, helping to prevent the model from becoming biased towards the majority classes.

Cross-Entropy[1] and SupCon Loss: In addition to Focal Loss[5], the experiment setup also utilizes cross-entropy[1] and SupCon loss. The latter is particularly used in the contrastive learning[2] setup, indicating a nuanced approach to balancing between learning distinctive features and achieving accurate classification.

Evaluation Metrics

Detailed Performance Analysis **Accuracy and Loss Metrics**: During training, both accuracy and loss are meticulously tracked, offering insights into the model's learning progress and effectiveness at minimizing classification error.

Confusion Matrix and Additional Metrics: The evaluation extends beyond simple accuracy,

employing confusion matrices, precision 6, recall 7, F1 8 scores, and potentially other metrics to provide a comprehensive view of model performance across all classes, which is crucial in the context of imbalanced datasets.

Metrics

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

Experiment Insights

Dynamic Experimentation Framework: The provided code establishes a dynamic framework for experimenting with different aspects of model training and evaluation. By adjusting parameters such as the imbalance ratio, block types in ResNet[4], loss function weights, and learning rate schedules, researchers can explore a vast space of configurations to identify optimal setups for specific challenges.

Class Imbalance Focus: A significant focus of the experimentation is on addressing class imbalance, a common challenge in real-world datasets. Through the use of imbalanced datasets, Focal Loss[5], and contrastive learning[2] strategies, the experiments aim to develop models that perform well across both majority and minority classes, ensuring fairness and robustness in model predictions.

6.2 CIFAR10

6.2.1 Resnet18

6.2.2 Analysis of Results

- **Precision and Recall**: The SupCon + CE method outperforms both CE and Focal Loss[5] in terms of precision and recall. This indicates that the hybrid method is not only more accurate in its positive

ResNet18[4] CIFAR10						
Method	Learning Rate	Batch Size	Precision	Recall	F1 Score	Accuracy
CE	0.1	32	0.748	0.722	0.723	0.722
Focal	0.1	32	0.722	0.709	0.701	0.709
SupCon + CE	0.1	32	**0.788**	**0.769**	**0.770**	**0.769**

Table 1: ResNet18[4] on CIFAR10

predictions (precision) but also more comprehensive in identifying all relevant instances (recall) seen in Table 1.

- **F1 Score**: Similarly, SupCon + CE achieves the highest F1 Score, which is a balanced measure that considers both precision and recall. This suggests that the method is effective in maintaining a balance between the precision and recall, making it superior for scenarios where both metrics are crucial seen in Table 1.

- **Accuracy**: Consistent with the other metrics, SupCon + CE also leads in accuracy. This demonstrates the overall effectiveness of combining contrastive learning[2] with traditional supervised learning for image classification tasks, see Table seen in Table 1.

The superior performance of the SupCon + CE method underscores the value of leveraging contrastive learning[2] in conjunction with cross-entropy loss[1], especially in a dataset with diverse and potentially imbalanced classes like CIFAR10. This approach not only enhances the model's discriminative capabilities but also its generalization power across different classes, see Table 1.

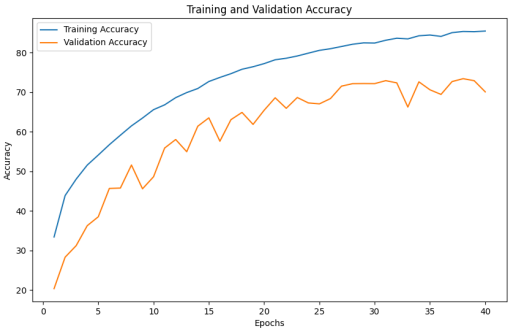

Figure 5: ResNet18[4] CE Accuracy Curve

Figure 6: ResNet18 CE Loss Curve

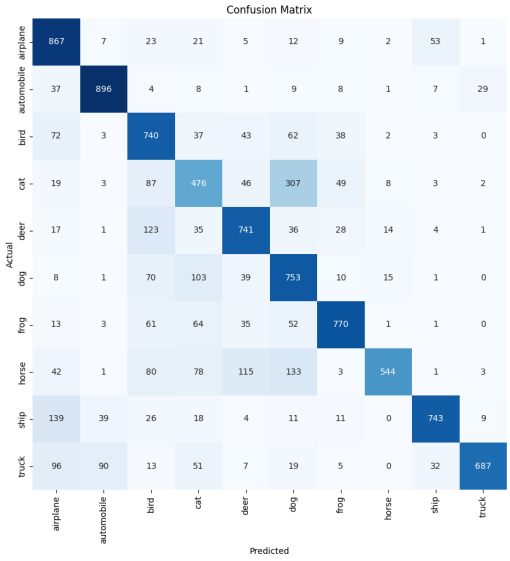

Figure 7: ResNet18 CE Confusion Matrix

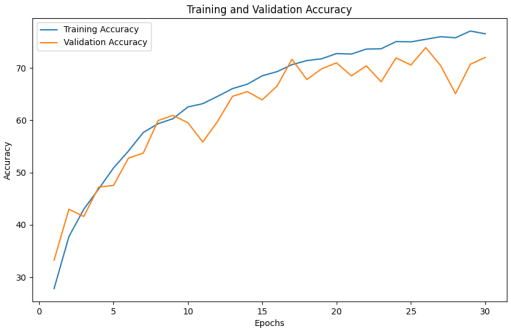

Figure 8: ResNet18 Focal Accuracy Curve

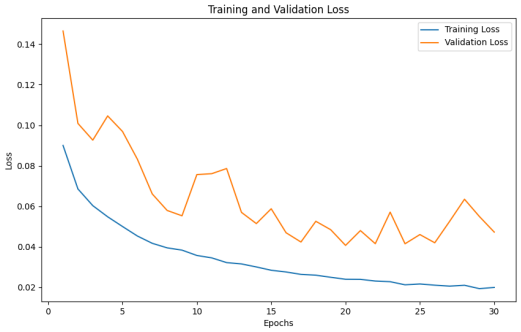

Figure 9: ResNet18 Focal Loss Curve

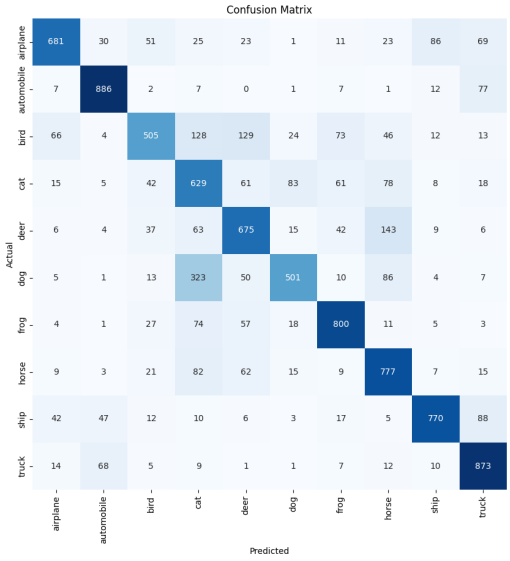

Figure 10: ResNet18 Focal Confusion Matrix

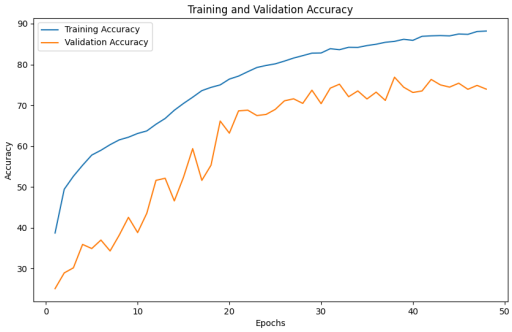

Figure 11: ResNet18 SupCon+CE Accuracy Curve

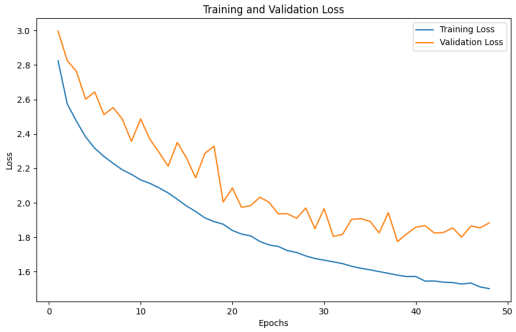

Figure 12: ResNet18 SupCon+CE Loss Curve

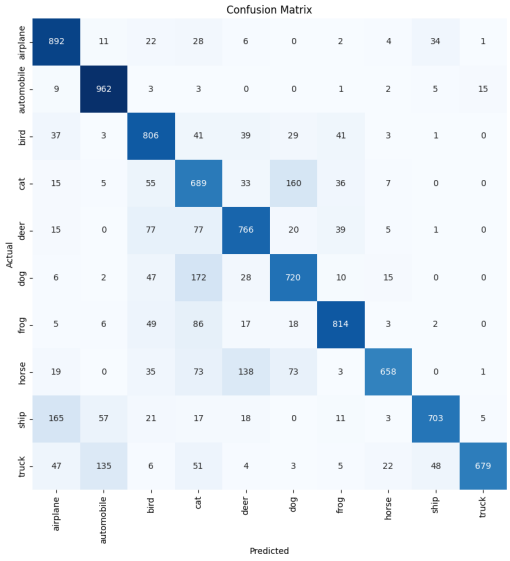

Figure 13: ResNet18 Focal Confusion Matrix

6.2.3 Visual Analysis

6.2.4 CE

Accuracy Curve Analysis

The accuracy curve provided indicates a steady improvement in training accuracy over epochs, which suggests that the model is learning effectively from the training data. However, there is a noticeable gap between training and validation accuracy. This could be due to the model learning features that are very specific to the training set, which do not generalize well to unseen data, see Figure 50.

Loss Curve Analysis

The loss curve shows a consistent decrease in training loss, indicating good convergence of the

model. However, similar to the accuracy curve, the validation loss appears to be higher than the training loss and exhibits some volatility. This oscillation in validation loss could be a sign of the model's struggle to generalize, as it indicates fluctuations in the model's performance on the validation set, see Figure 51.

Confusion Matrix Analysis

The confusion matrix gives a more detailed look at the model's performance across different classes. From the matrix, we can see that the model performs well on some classes (e.g., 'automobile' and 'ship'), but seems to struggle with others (e.g., 'cat' and 'dog'). This kind of misclassification between certain classes may stem from similarities in the features of these classes that the model finds challenging to distinguish. Additionally, it might also point towards a limitation in the representational power of the model to capture the nuances between such similar classes, see Figure 7.

Overall Analysis

Considering the results from the table, the model using CE loss performs reasonably well with an accuracy of 72.2%. However, this could potentially be improved. The slightly lower precision compared to recall suggests that the model is making more false positive errors, which in turn affects the F1 score. The F1 score being close to the accuracy indicates that the model's performance is relatively balanced across classes.

6.2.5 Focal

Accuracy Curve Analysis

The accuracy curve shows that the training accuracy steadily increases and plateaus around 70%, which suggests the model is learning from the training dataset. However, the validation accuracy fluctuates more compared to the training accuracy and does not reach the same level, peaking around 60%. This discrepancy indicates that the model may be learning specific patterns not generalizable to the validation set, which is a common challenge when using the Focal loss[5] function due to its focus on harder-to-classify examples, see Figure 52.

Loss Curve Analysis

The loss curve presents a decreasing trend in training loss, demonstrating that the model is learning and improving its predictions over epochs. However, the validation loss shows higher variability and doesn't decrease as smoothly. The oscillations in the validation loss suggest that the model's performance on the validation set is not consistent and may benefit from further tuning of the Focal loss[5] hyperparameters to

stabilize learning, see Figure 53.

Confusion Matrix Analysis

The confusion matrix for the Focal[5] method shows that certain classes, such as 'ship' and 'truck', are classified with high accuracy. However, there are classes like 'cat' and 'dog' where the model is less accurate, which can be attributed to the Focal loss[5] emphasizing the learning of more complex patterns, possibly at the expense of simpler but critical features necessary for distinguishing between similar classes, see Figure 10.

Overall Analysis

Considering the CIFAR10 table results for the Focal method[5], we see slightly reduced performance across Precision, Recall, F1 Score, and Accuracy compared to the CE method. This might be due to Focal loss[5] focusing on misclassified or difficult examples, which can sometimes lead to neglecting the 'easier' examples that are still important for overall accuracy. It's a balance that needs to be carefully managed to avoid biasing the model too much toward the difficult cases, , see Figure 1.

6.2.6 SupCon+CE

Accuracy Curve Analysis

The accuracy curve for the SupCon+CE method indicates a stable increase in training accuracy, reaching close to 90%. This high level of training accuracy suggests that the model is learning effectively and can capture the features necessary to classify the training images. The validation accuracy also shows an increasing trend but plateaus around 70%. The validation accuracy is relatively stable, which is a good sign of the model's generalization capabilities, see Figure 36.

Loss Curve Analysis

The loss curve for the SupCon+CE method depicts a sharp decline in training loss, which levels off as the epochs increase, suggesting that the model is converging well. The validation loss decreases alongside the training loss but exhibits some fluctuations in the later epochs. These fluctuations could indicate that the model is sensitive to the initialization and variability in the training data presented in each epoch, see Figure37.

Confusion Matrix Analysis

The confusion matrix for the SupCon+CE method shows a strong diagonal, indicating that most classes are correctly classified. There are, however, some misclassifications, such as between 'cat' and 'dog,'

which are common due to their visual similarities. The relatively high numbers on the matrix's diagonal line suggest that the model, trained with the SupCon+CE method, is quite effective in distinguishing between most of the CIFAR10 classes, see Figure 13.

Overall Analysis

The SupCon+CE method, which combines the strengths of Supervised Contrastive Learning[2] with Cross-Entropy[1], appears to have resulted in a model that not only performs well on the training set but also has a decent generalization to the validation set. The high training accuracy and more stable validation accuracy, as compared to the Focal method[5], suggest that SupCon+CE is a strong method for this dataset.

6.2.7 Resnet34

ResNet34 CIFAR10						
Method	Learning Rate	Batch Size	Precision	Recall	F1 Score	Accuracy
CE	0.1	32	0.777	0.749	0.753	0.749
Focal	0.1	32	0.736	0.731	0.726	0.731
SupCon + CE	0.1	32	**0.813**	**0.791**	**0.792**	**0.791**

Table 2: ResNet34 on CIFAR10

6.2.8 Analysis of Results

- **Precision**: The SupCon + CE method outperforms CE and Focal Loss[5] with the highest precision of 0.813. This indicates that when a class is predicted, it is correct more often for the SupCon + CE method than the others, see Table 2.
- **Recall**: Similarly, the SupCon + CE method achieves the highest recall, indicating that it is better at identifying all relevant instances within a class, see Table 2.
- **F1 Score**: The F1 Score is the harmonic mean of precision and recall and is particularly important in situations where a balance between precision and recall is needed. The SupCon + CE method leads in this metric as well, suggesting it is the most balanced approach among those tested, see Table 2.
- **Accuracy**: Reflecting the trends in precision and F1 Score, the SupCon + CE method also has the highest overall accuracy, demonstrating its effectiveness across all classes in CIFAR10, see Table 2.

The results from the ResNet34[4] model trained on the CIFAR10 dataset highlight the effectiveness

of the SupCon + CE method, which has shown superior performance across all metrics when compared to CE and Focal Loss[5]. The integration of Supervised Contrastive Learning[2] with Cross-Entropy Loss[1] has proven beneficial in creating a model that not only predicts with higher precision but also recalls a larger proportion of relevant instances across the classes.

The leading F1 Score for the SupCon + CE method indicates a balanced model that does not excessively favor precision over recall or vice versa, which is essential for a dataset with a diverse set of classes such as CIFAR10. Furthermore, the high accuracy achieved by the SupCon + CE method underscores its overall effectiveness and potential as a robust approach for image classification tasks. These results advocate for the adoption of hybrid training strategies, particularly in complex datasets, to improve the precision and reliability of predictive models.

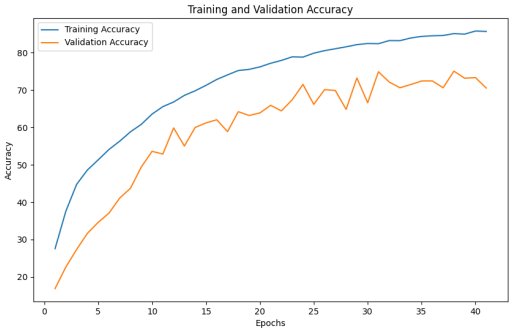

Figure 14: ResNet34 CE Accuracy Curve

Figure 15: ResNet34 CE Loss Curve

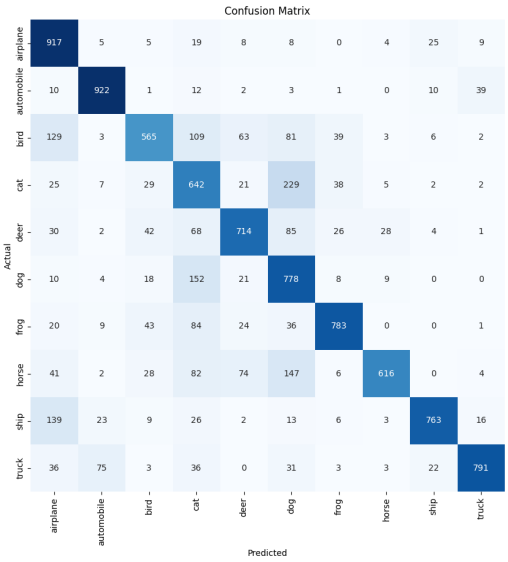

Figure 16: ResNet34 CE Confusion Matrix

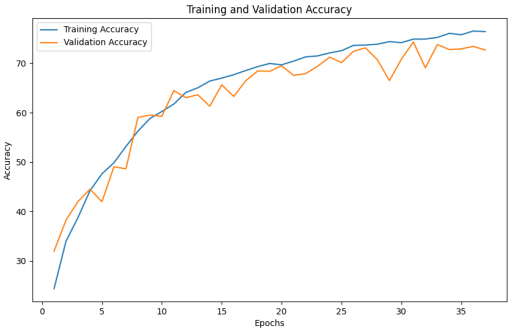

Figure 17: ResNet34 Focal Accuracy Curve

Figure 18: ResNet34 Focal Loss Curve

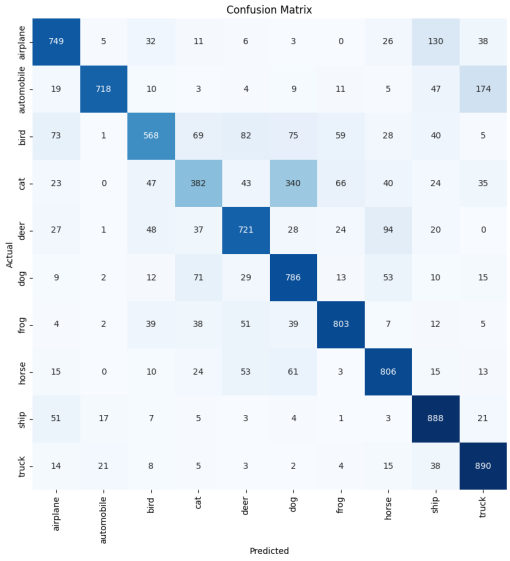

Figure 19: ResNet34 Focal Confusion Matrix

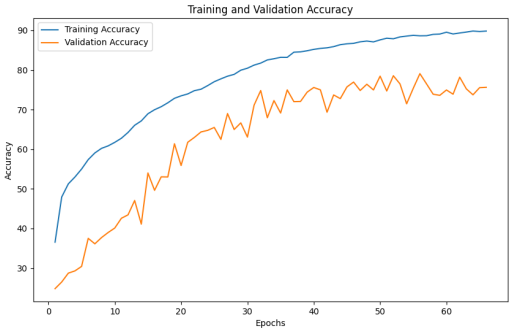

Figure 20: ResNet34 SupCon+CE Accuracy Curve

Figure 21: ResNet34 SupCon+CE Loss Curve

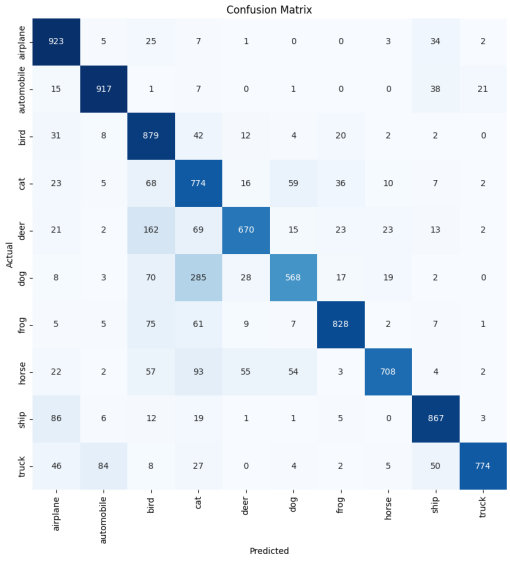

Figure 22: ResNet34 SupCon+CE Confusion Matrix

6.2.9 Visual Analysis

6.2.10 CE

Accuracy Curve Analysis

The provided accuracy curve shows that the training accuracy steadily increases throughout the epochs, indicating that the model is learning effectively and continuously improving its ability to correctly classify the training data. The validation accuracy also increases but seems to plateau around 70%. This plateau could imply that the model has reached its capacity for generalization with the current setup. see Figure 14.

Loss Curve Analysis

The loss curve exhibits a decreasing trend in training loss, which is a good sign of the model's ability to minimize the error over time. The validation loss decreases alongside the training loss but shows more fluctuation, which is typical in validation metrics as the model is tested against data it hasn't seen before. see Figure 15.

Confusion Matrix Analysis

The confusion matrix gives a detailed account of the model's performance across all classes. High values along the diagonal indicate correct classifications, with particularly strong performance for classes like 'automobile' and 'ship'. However, there are notable confusions between certain classes such as 'cat' and 'dog', and 'bird' and 'plane', which may share similar features and thus are harder for the model to distinguish, see Figure 16.

Overall Analysis

With the CE method, the ResNet34[4] model achieves an accuracy of 74.9%, along with precision, recall, and F1 scores in the same range. These results are fairly balanced, suggesting that the model has a consistent performance across different classes without significant bias.

6.2.11 Focal

Accuracy Curve Analysis The accuracy curve for the Focal method[5] shows that the training accuracy improves consistently as the epochs progress, which indicates effective learning from the training data. However, the validation accuracy, after an initial increase, shows some fluctuation and then levels off, suggesting that the model may be having difficulties generalizing to the validation data beyond a certain point. see Figure 17.

Loss Curve Analysis

The loss curve displays a continuous decline in training loss, which is expected as the model optimizes its parameters. The validation loss decreases initially but then starts to fluctuate, indicating that the model's improvements on the training data are not consistently reflected in its performance on the validation set. This could be a result of the Focal loss[5] function focusing the model's learning on hard examples, potentially at the expense of not adequately learning simpler but still informative features, see Figure 18.

Confusion Matrix Analysis

The confusion matrix provides insights into the model's performance on a class-by-class basis. For the Focal method[5], certain classes such as 'ship' and 'truck' show high correct classification rates, which

is positive. However, there are noticeable confusions between classes like 'cat' and 'dog', as well as 'bird' and 'deer', which may be due to these classes sharing similar features that the model, guided by the Focal loss[5], finds difficult to distinguish, see Figure 19.

Overall Analysis

In terms of numerical performance, the Focal method[5] achieves an accuracy of 73.1%, with precision, recall, and F1 Score slightly lower compared to the CE method. This suggests that while the Focal loss[5] helps in focusing on difficult examples, it may not be optimizing the overall balance between precision and recall as effectively as the CE method in this case.

6.2.12 SupCon+CE

Accuracy Curve Analysis The accuracy curve indicates a significant difference between training and validation accuracy, with the training accuracy being much higher. This gap suggests the model is learning the training data well but may not be generalizing effectively to the validation data. see Figure 20.

Loss Curve Analysis

The loss curve shows a consistent decrease in training loss, indicating that the model is improving and learning from the training data over time. However, the validation loss seems to plateau and fluctuate, see Figure 21.

Confusion Matrix Analysis

The confusion matrix reveals the model's performance on individual classes. There is a strong diagonal, indicating correct classifications for most classes. Some confusion is present between similar classes such as 'cat' and 'dog', which is a common issue due to the visual similarities between these classes. The confusion matrix can be used to identify which classes might need further data augmentation or more nuanced feature learning, see Figure 22.

Overall Analysis

The SupCon+CE method has led to high training accuracy, but the validation accuracy indicates the model may have memorized the training data rather than learned generalizable features. This method typically aims to improve feature representation by learning more about the relationships between different classes.

6.2.13 Resnet50

ResNet50 CIFAR10						
Method	Learning Rate	Batch Size	Precision	Recall	F1 Score	Accuracy
CE	0.1	32	0.692	0.665	0.665	0.665
Focal	0.1	32	0.715	0.711	0.707	0.711
SupCon + CE	0.1	32	**0.784**	**0.771**	**0.771**	**0.771**

Table 3: ResNet50 on CIFAR10

6.2.14 Analysis of Results

- **Precision**: The highest precision is achieved with the SupCon + CE method at 78.4%, indicating that this approach is the most precise in classifying images correctly among the tested methods. In comparison, the Focal method[5] shows an improvement over the standard CE, with a precision of 71.5%, see Table 3.
- **Recall**: Similarly, SupCon + CE leads in recall with 77.1%, suggesting that it is better at identifying all relevant instances within the classes. The Focal method[5] also surpasses CE, which implies it is more effective in dealing with class imbalances present in the dataset, see Table 3.
- **F1 Score**: The F1 Score, which balances precision and recall, is again the highest for SupCon + CE at 77.1%. This is indicative of a well-rounded model that does not excessively trade off precision for recall or vice versa. The Focal method's[5] F1 Score shows it has a better balance than CE but still falls short of the hybrid method, see Table 3.
- **Accuracy**: For overall accuracy, SupCon + CE demonstrates superior performance at 77.1%, significantly higher than both the CE and Focal methods[5]. This reinforces the effectiveness of combining contrastive learning[2] with cross-entropy[1] in achieving a more accurate model on CIFAR10, see Table 3

The comparative analysis of the ResNet50[4] model on the CIFAR10 dataset clearly illustrates the superiority of the SupCon + CE method over the standard CE and Focal methods[5] across all evaluated metrics. By effectively leveraging the strengths of supervised contrastive learning[2] combined with cross-entropy loss[1], the model not only improves in distinguishing between classes but also in generalizing from the training data to unseen data. These results suggest that future works should consider hybrid approaches

like SupCon + CE, especially in scenarios where complex data distributions are present, to enhance model performance and reliability.

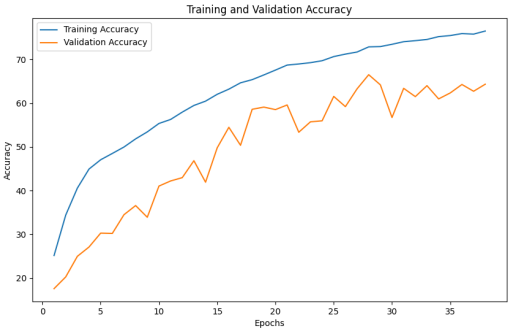

Figure 23: ResNet50 CE Accuracy Curve

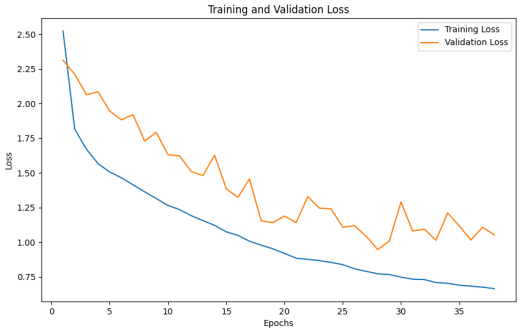

Figure 24: ResNet50 CE Loss Curve

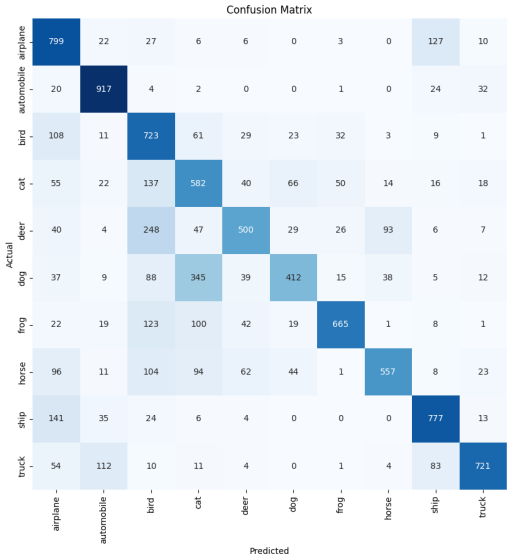

Figure 25: ResNet50 CE Confusion Matrix

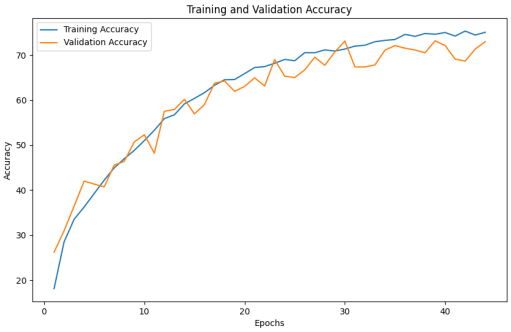

Figure 26: ResNet50 Focal Accuracy Curve

Figure 27: ResNet50 Focal Loss Curve

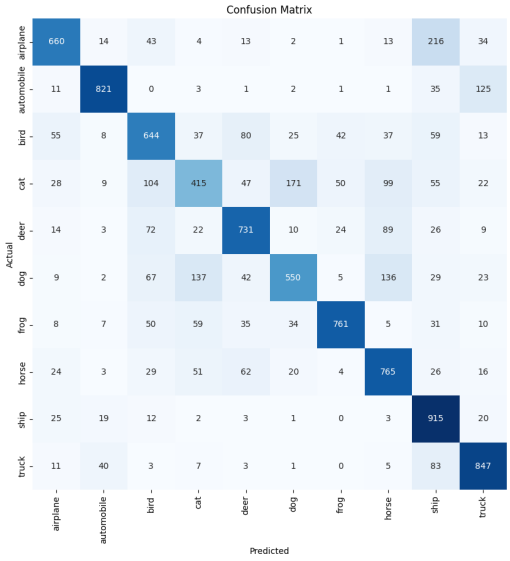

Figure 28: ResNet50 Focal Confusion Matrix

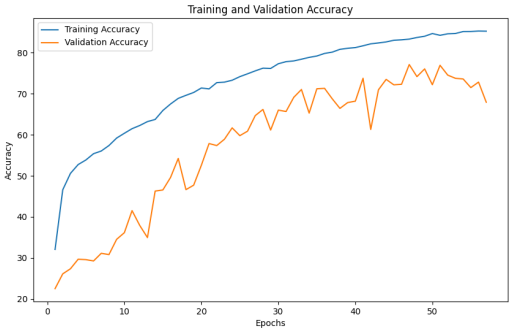

Figure 29: ResNet50 SupCon+CE Accuracy Curve

Figure 30: ResNet50 SupCon+CE Loss Curve

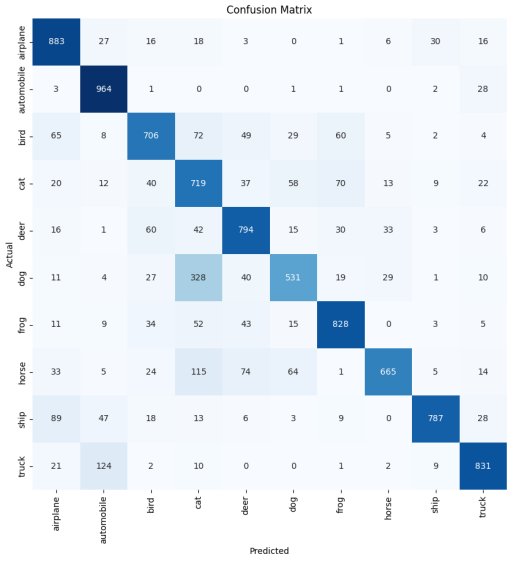

Figure 31: ResNet50 SupCon+CE Confusion Matrix

6.2.15 Visual Analysis

6.2.16 CE

Accuracy Curve Analysis The accuracy curve for the CE method reveals a consistent increase in training accuracy, reaching above 70%. The validation accuracy initially follows the training accuracy but begins to plateau and shows variability, see Figure 23.

Loss Curve Analysis The training loss curve demonstrates a steady decline, indicating effective learning and optimization of model parameters. In contrast, the validation loss declines but with noticeable fluctuations, particularly in later epochs. This behavior often signifies that while the model is becoming better at fitting the training data, it is not consistently improving its generalization to the validation set,

see Figure 24.

Confusion Matrix Analysis The confusion matrix displays a strong diagonal, indicating a high number of correct predictions across most classes. However, there are significant misclassifications for certain classes, suggesting that the model may be confusing features between these classes. It is particularly evident in classes where the visual distinction is subtler, which can lead to higher confusion rates, see Figure 25.

Overall Analysis The results from the ResNet50[4] model using the CE method on CIFAR10 show that while the model achieves a relatively high training accuracy, there is a notable discrepancy with the validation accuracy. This discrepancy points towards overfitting, which is further supported by the fluctuations in the validation loss. The confusion matrix corroborates this by showing certain classes where the model performance is not optimal. To enhance the model's generalization ability, strategies such as data augmentation, regularization, and hyperparameter tuning should be considered.

6.2.17 Focal

Accuracy Curve Analysis The accuracy curve depicts a consistent increase in the training accuracy, signifying that the model is learning and improving from the training data. The validation accuracy, however, shows a more volatile progression with several peaks and troughs, see Figure 26.

Loss Curve Analysis The training loss demonstrates a steady decrease, which is an expected trend as the model is optimizing. The validation loss, in contrast, decreases but with fluctuations and a less pronounced decline. This variability might suggest that while the model is becoming adept at fitting to the training data, it is not steadily improving on the validation set, see Figure 27.

Confusion Matrix Analysis The confusion matrix shows the number of correct and incorrect predictions made by the model. It indicates that certain classes are well-identified, whereas others, particularly those with similar features, are often confused. This suggests that while Focal loss[5] helps in focusing training on hard examples, it may not be as effective in distinguishing between certain classes, see Figure 28.

Overall Analysis Overall, the Focal method[5] shows an improvement in handling the dataset's class imbalance compared to the standard CE method, as indicated by the higher accuracy and lower loss. However, the accuracy and loss curves suggest that the model might benefit from techniques to improve its generalization capabilities, such as regularization and data augmentation. Furthermore, the confusion matrix points out specific classes where the model's performance could be enhanced, perhaps by fine-tuning the focal[5] parameters or employing targeted data augmentation strategies.

6.2.18 SupCon+CE

Accuracy Curve Analysis

The accuracy curve for the SupCon+CE (Supervised Contrastive Learning[2] combined with Cross-Entropy)[1] model applied to ResNet50[4] on CIFAR-10 dataset shows a clear trend of increasing training accuracy over the epochs. The training accuracy starts at around 30% and gradually increases to just over 80% by the 50th epoch. This indicates that the model is effectively learning from the training data over time. However, the validation accuracy is more volatile and does not show a similar smooth increase. It starts at around the same level as the training accuracy but only reaches a peak of about 70% before declining and ending at around 65%.See Figure 29.

Loss Curve Analysis

The loss curve complements the accuracy curve, with the training loss decreasing steadily from around 2.8 to just under 1.6. This decrease in loss indicates that the model is getting better at making predictions on the training data. The validation loss, on the other hand, decreases less smoothly and seems to plateau around the 2.0 mark. see Figure 30.

Confusion Matrix Analysis

The confusion matrix provides insight into the model's performance on individual classes. The diagonal values represent the number of correct predictions for each class, with higher values indicating better performance. The model performs best in classifying 'automobile', 'ship', and 'truck', with correct predictions of 964, 787, and 831 respectively. The classes 'bird', 'cat', and 'dog' have the most confusion, with the model often misclassifying between these three categories. The 'frog' class also sees a high number of correct predictions (828), but there is notable confusion with the 'cat' class, see Figure 31.

Overall Analysis

Overall, the SupCon+CE model trained on the CIFAR-10 dataset using ResNet50[4] architecture shows a promising increase in training accuracy over time. The model performs very well on certain classes such as 'automobile', 'ship', and 'truck', but struggles with 'bird', 'cat', and 'dog', which may be due to similarities between these classes causing confusion.

6.3 CIFAR100

6.3.1 Resnet18

ResNet18 CIFAR100						
Method	Learning Rate	Batch Size	Precision	Recall	F1 Score	Accuracy
CE	0.1	32	**0.475**	0.388	**0.378**	0.388
Focal	0.1	32	0.384	0.374	0.351	0.374
SupCon + CE	0.1	32	0.368	**0.393**	0.328	**0.393**

Table 4: ResNet18 on CIFAR100

6.3.2 Analysis of Results

- **Precision**: The CE method exhibits the highest precision with a score of 0.475, indicating that it has the highest proportion of true positive identifications relative to the number of true positives and false positives combined. This suggests that the CE method is relatively reliable when it classifies an instance as positive, see Table 4.
- **Recall**: SupCon + CE achieves the best recall score of 0.393, showing that it is more effective than the other methods at identifying all relevant instances. This method's higher recall indicates a better capability to find all the positive samples, but not necessarily correctly identify only the positives, as evidenced by its precision, see Table 4.
- **F1 Score**: The CE method has the highest F1 score at 0.378, which suggests a balanced performance between precision and recall. Despite its lower recall compared to SupCon + CE, the CE method's higher precision contributes to a better F1 score, showing a balanced trade-off between the two metrics, see Table 4.
- **Accuracy**: Both CE and SupCon + CE share the highest accuracy score of 0.393. This indicates that, overall, they classify the correct labels for a given input more frequently than the Focal method[5], considering all classes, see Table 4.

Conclusion

In summary, the CE method appears to be the most effective when considering precision and F1 score, which implies a better overall balance of false positives and false negatives in classification on

CIFAR100 using ResNet18[4]. SupCon + CE, while having the highest recall, does not achieve the same level of precision or F1 score, indicating a potential over-classification of positives. The Focal method[5] lags behind in all performance metrics, suggesting that the balancing mechanism of Focal loss[5] is not as beneficial for this dataset and model as the other methods. Given that CE and SupCon + CE tie in accuracy, the choice between them may depend on the specific needs of the application: CE for cases where false positives are more costly, and SupCon + CE where missing out on true positives is a greater concern.

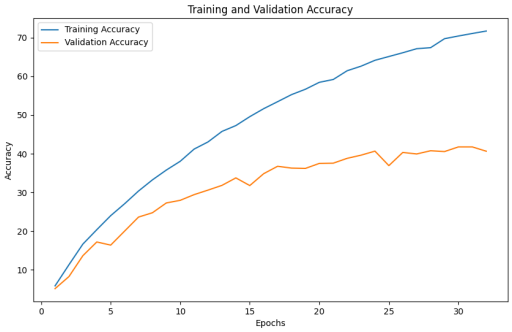

Figure 32: ResNet18 CE Accuracy Curve

Figure 33: ResNet18 CE Loss Curve

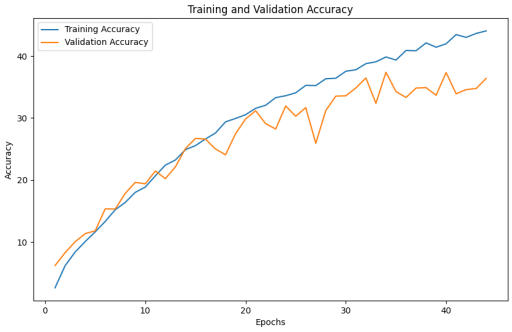

Figure 34: ResNet18 Focal Accuracy Curve

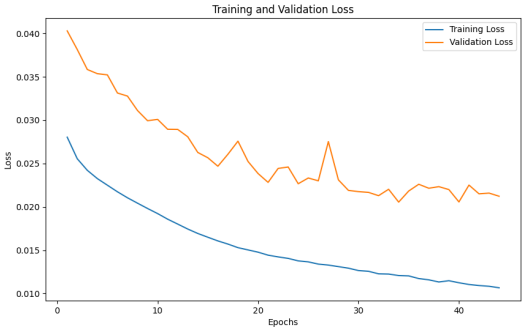

Figure 35: ResNet18 Focal Loss Curve

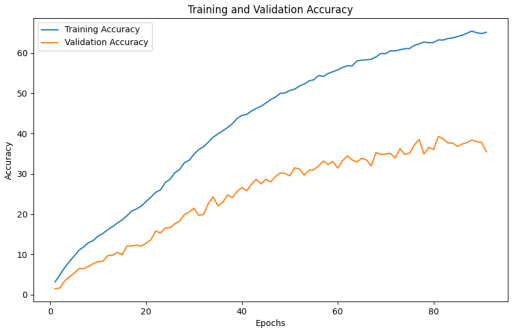

Figure 36: ResNet18 SupCon+CE Accuracy Curve

Figure 37: ResNet18 SupCon+CE Loss Curve

6.3.3 Visual Analysis

6.3.4 CE

Accuracy Curve Analysis

The accuracy curve for the Cross-Entropy (CE)[1] loss function when applied to a ResNet18[4] model trained on the CIFAR100 dataset shows a continuous improvement in training accuracy over 30 epochs. The training accuracy starts at around 10% and steadily increases to approximately 70%. This consistent increase suggests that the model is effectively learning and improving its ability to classify the 100 different classes in the CIFAR100 dataset. The validation accuracy, however, starts at a similar point but only increases to about 30%, and from epoch 10 onwards, it plateaus with slight fluctuations. See Figure 32.

Loss Curve Analysis

The loss curve exhibits a decreasing trend in training loss, which drops from around 4.5 to below 1.0, suggesting that the model is becoming more confident in its predictions on the training set as the epochs progress. However, the validation loss decreases only until it reaches approximately 2.5 by the 5th epoch and then fluctuates around this value for the remainder of the epochs. See Figure 33.

Overall Analysis

The analysis of the accuracy and loss curves for the CE method with ResNet18[4] on CIFAR100 points towards a model that is learning well. The model's training accuracy is high, and training loss is low, yet it fails to generalize these results effectively to the validation data, as shown by the lower validation accuracy and higher validation loss.

6.3.5 Focal

Accuracy Curve Analysis

The accuracy curve for the Focal loss[5] function applied to a ResNet18[4] model on the CIFAR100 dataset shows a positive trend in both training and validation accuracy over 50 epochs. The training accuracy starts at approximately 10% and exhibits a steady increase, plateauing around 40% towards the later epochs. The validation accuracy, although following a similar upward trend, demonstrates greater variability and ends at around 30%. The convergence between training and validation accuracy is not apparent, see Figure 35.

Loss Curve Analysis

The loss curve reveals that the training loss decreases significantly from approximately 0.040 to

just below 0.010, indicating improved model performance on the training data as learning progresses. Conversely, the validation loss decreases from around 0.035 to about 0.025 before plateauing and then exhibits fluctuations without a clear downward trend. This pattern in the validation loss suggests that the model's ability to generalize to unseen data may not be improving significantly after a certain point, see Figure 35.

Overall Analysis

Overall, the Focal loss[5] function with ResNet18[4] on CIFAR100 leads to an increase in training accuracy and a decrease in training loss, indicative of learning. However, the model's performance on validation data does not mirror this improvement, as evidenced by the lower validation accuracy and higher validation loss. The discrepancy between training and validation metrics indicates that the model may be overfitting the training data, and the Focal loss[5] function's intrinsic focus on harder, misclassified examples does not seem to close this gap.

6.3.6 SupCon+CE

Accuracy Curve Analysis

The accuracy curve for the SupCon+CE method applied to ResNet18[4] on the CIFAR100 dataset shows a training accuracy that starts at approximately 10% and steadily climbs to surpass 60% by the end of 100 epochs. This gradual increase indicates that the model is continuously learning and improving its ability to correctly classify images from the dataset. The validation accuracy begins at a similar level but only reaches about 25%. Despite some fluctuations, it maintains a consistent upward trajectory, though it remains significantly lower than the training accuracy. The persistent gap between training and validation accuracy suggests that while the model is learning the training data effectively, it may not be generalizing as well to new, unseen data, see Figure 36

Loss Curve Analysis

The training loss curve starts at around 3.5 and decreases to just over 1.0, which corroborates the positive learning trend shown in the accuracy curve. The model is becoming more confident and making fewer errors on the training set over time. On the other hand, the validation loss decreases alongside the training loss until approximately the 20th epoch, where it begins to plateau around 1.5. The plateau and slight upward trend in the later epochs may indicate that the model's ability to generalize is not improving further, see Figure 37.

Overall Analysis

Overall, the SupCon+CE approach shows a model that is capable of learning from the training data with a steady improvement in accuracy and decrease in loss. However, the less pronounced improvement on the validation set suggests that the model's generalization to unseen data is limited. To counteract this, strategies such as implementing regularization techniques, fine-tuning the model's hyperparameters, or using more advanced data augmentation methods could be beneficial. Moreover, considering modifications to the SupCon+CE approach, like adjusting the weight given to the supervised contrastive loss[2] component, could potentially enhance the model's generalization performance.

6.3.7 Resnet34

ResNet34 CIFAR100						
Method	Learning Rate	Batch Size	Precision	Recall	F1 Score	Accuracy
CE	0.1	32	**0.468**	0.404	0.386	0.404
Focal	0.1	32	0.450	**0.434**	**0.424**	**0.434**
SupCon + CE	0.1	32	0.376	0.386	0.327	0.386

Table 5: ResNet34 on CIFAR100

6.3.8 Analysis of Results

- **Precision**: In terms of precision, which measures the accuracy of positive predictions, the CE method outperforms the other methods with a score of 0.468. This indicates that when the CE method predicts an instance as positive, it is more likely to be correct compared to the other methods, see Table 5.
- **Recall**: For recall, which assesses the model's ability to identify all actual positive instances, Focal loss[5] leads with a score of 0.434. This suggests that Focal loss[5] is more effective at capturing the majority of positive instances in the dataset, see Table 5.
- **F1 Score**: The F1 score is the harmonic mean of precision and recall and is used to measure a test's accuracy. The Focal method[5] achieves the highest F1 score at 0.424, reflecting a balanced performance between precision and recall, indicating its robustness in handling both false positives and false negatives, see Table 5.
- **Accuracy**: Regarding overall accuracy, which reflects the proportion of true results (both true positives and true negatives) among the total number of cases examined, the Focal method[5] again ranks highest

with a score of 0.434. This means it correctly labels more instances across all classes than the other methods, see Table 5.

Conclusion

The comparison of CE, Focal, and SupCon + CE methods for ResNet34[4] on CIFAR100 shows that while CE has the highest precision, it is the Focal method[5] that leads in recall, F1 score, and accuracy. The Focal method's[5] superior recall and F1 score suggest it is particularly effective in a dataset like CIFAR100, where there may be a high class imbalance, and the cost of missing a positive instance is significant. The SupCon + CE method, while innovative in combining supervised contrastive learning[2] with cross-entropy[1], does not perform as well on these metrics, which could indicate that the balance between the two loss functions needs to be optimized for this specific task. Given the complexity and variability of the CIFAR100 dataset, the Focal method's[5] ability to focus on harder-to-classify examples and reduce the weight of well-classified examples seems to give it an edge in overall performance.

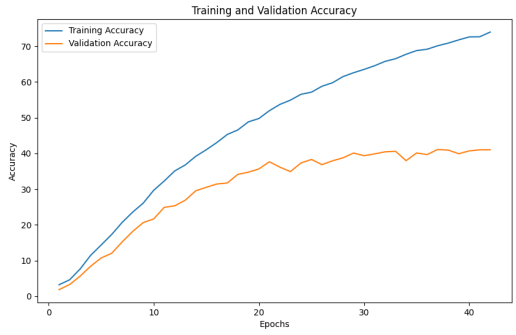

Figure 38: ResNet34 CE Accuracy Curve

Figure 39: ResNet34 CE Loss Curve

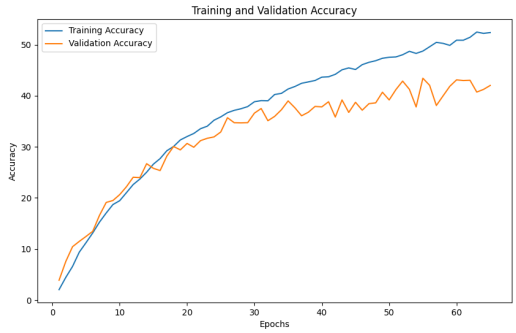

Figure 40: ResNet34 Focal Accuracy Curve

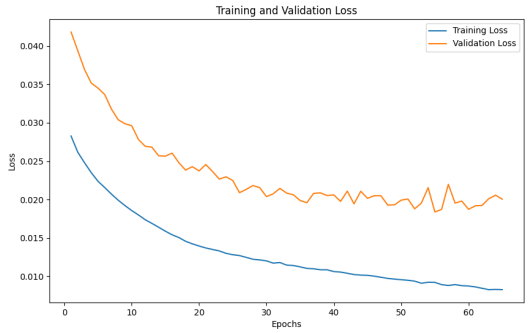

Figure 41: ResNet34 Focal Loss Curve

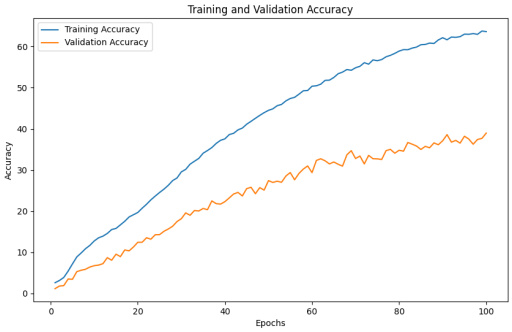

Figure 42: ResNet34 SupCon+CE Accuracy Curve

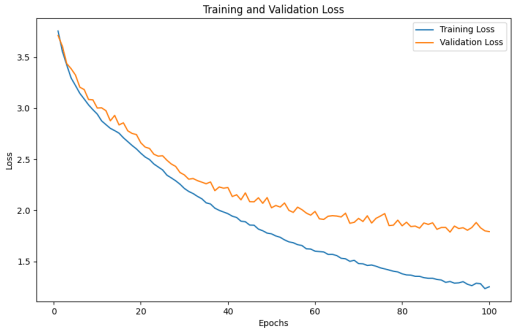

Figure 43: ResNet34 SupCon+CE Loss Curve

6.3.9 Visual Analysis

6.3.10 CE

Accuracy Curve Analysis

For the CE (Cross-Entropy)[1] method on a ResNet34[4] model trained on CIFAR100, the training accuracy curve starts at around 10% and exhibits a stable and consistent increase throughout the 50 epochs, reaching above 60%. This steady progression indicates that the model is learning effectively from the training data. However, the validation accuracy starts at a similar level but plateaus around 30%, with minor fluctuations throughout the training process. See Figure 38.

Loss Curve Analysis

The training loss curve demonstrates a sharp decline from around 4.5 to below 1.5, indicating that the model is getting progressively better at classifying the training images correctly. In contrast, the validation loss decreases alongside the training loss initially but begins to plateau around the 2.5 mark, showing little improvement as the epochs increase. see Figure 41.

Overall Analysis

Overall, the CE approach for ResNet34[4] on CIFAR100 indicates a model that is capable of learning from the training data but shows signs of disparities between training and validation performance. The model's ability to generalize to new data is not as strong as its ability to learn from the training data.

6.3.11 Focal

Accuracy Curve Analysis

The accuracy curve for the Focal loss[5] function applied to a ResNet34[4] architecture on the CIFAR100 dataset shows an upward trajectory for training accuracy, beginning at around 10% and reaching approximately 50% by the 70th epoch. This steady climb suggests the model is effectively learning from the training data. However, the validation accuracy starts at the same level but plateaus around 40% much earlier, around the 30th epoch, with some variability afterwards. See Figure 40.

Loss Curve Analysis

The training loss decreases substantially from about 0.040 to below 0.010, suggesting that the model is becoming more accurate in its predictions over time. In contrast, the validation loss declines alongside the training loss initially but then levels off around 0.015 and displays some fluctuations for the remainder of the training. This pattern of validation loss indicates that the model's improvement on the validation set is not as pronounced as it is on the training set, see Figure 40.

Overall Analysis

Overall, the use of Focal loss[5] with ResNet34[4] on the CIFAR100 dataset shows a model capable of learning and improving its predictions on the training set but with limitations in generalizing these improvements to the validation set. The persistent gap between training and validation accuracy, along with the plateau in validation loss. To address this, techniques such as data augmentation, introduction of regularization (like dropout or weight decay), and hyperparameter tuning could be explored to enhance the model's generalization capabilities. Additionally, experimenting with the gamma parameter of the Focal loss[5], which focuses the training on hard-to-classify examples, might help in improving the performance on the validation set.

6.3.12 SupCon+CE

Accuracy Curve Analysis

The accuracy curve for the SupCon+CE method when applied to ResNet34[4] on CIFAR100 shows that the training accuracy begins at around 10% and follows a steady upward trend, reaching just over 60% by the end of 100 epochs. This indicates a good learning progression. The validation accuracy also starts at about 10% but plateaus near 35% after about 60 epochs, demonstrating a significant and sustained gap with the training accuracy. See Figure 42.

Loss Curve Analysis

The loss curve reflects a consistent decrease in training loss from around 3.5 to below 1.5, which is

indicative of the model learning and improving its prediction accuracy on the training data. The validation loss, while also decreasing, shows a more gradual reduction and begins to plateau around 1.5. It does not achieve the low levels seen in the training loss, see Figure 43.

Overall Analysis

Overall, the SupCon+CE training for ResNet34[4] on CIFAR100 illustrates that the model is capable of learning from the training data, as shown by the increase in training accuracy and the decrease in training loss. However, the discrepancy between the training and validation results, with validation metrics plateauing at a much lower level than the training metrics, indicates a potential issue with overfitting. This suggests that while the model is fitting the training data well, it is not as capable when it comes to generalizing to new, unseen data. To improve the model's generalization, techniques such as data augmentation, regularization, and perhaps adjustment of the SupCon+CE loss function parameters could be considered.

6.3.13 Resnet50

ResNet50 CIFAR100						
Method	Learning Rate	Batch Size	Precision	Recall	F1 Score	Accuracy
CE	0.1	32	**0.468**	0.404	0.386	0.404
Focal	0.1	32	0.450	**0.434**	**0.424**	**0.434**
SupCon + CE	0.1	32	0.376	0.386	0.327	0.386

Table 6: ResNet50 on CIFAR100

6.3.14 Analysis of Results

- **Precision**: The CE method tops the precision metric with a score of 0.468, indicating that it has the highest likelihood of its positive predictions being correct when compared to the other methods, see Table 6.
- **Recall**: Focal loss[5] demonstrates the highest recall at 0.434, meaning it is more effective than the other methods at identifying all the relevant instances in the dataset, see Table 6.
- **F1 Score**: With regards to the F1 score, which balances precision and recall, the Focal method[5] again leads with a score of 0.424, suggesting that it has the most balanced performance between precision and recall, see Table 6.

- **Accuracy**: In terms of overall accuracy, the Focal method[5] stands out as the best performer with a score of 0.434, which indicates that it correctly classifies a higher percentage of instances across all classes, see Table 6.

Conclusion The performance metrics of ResNet50[4] on CIFAR100 reveal that the CE method has the highest precision, suggesting it is relatively more reliable in its positive classifications. The Focal loss[5] method, however, outperforms in recall, F1 score, and accuracy, indicating it is generally more effective across the board. This could be due to the Focal loss's[5] ability to focus more on difficult-to-classify examples, which seems to benefit the overall performance on a diverse and complex dataset like CIFAR100. The SupCon + CE method, while innovative in its approach to combine contrastive learning[2] with cross-entropy[1], does not seem to measure up to the Focal method[5] in this instance, possibly due to a less optimal balance between learning from hard negatives and classifying correctly. The findings suggest that for a dataset with a large number of classes and potential class imbalance, a method that can address class imbalance like Focal loss[5] might be more advantageous.

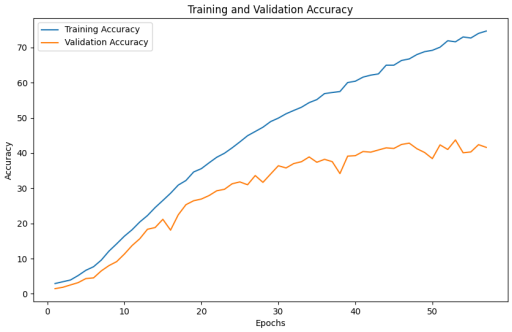

Figure 44: ResNet50 CE Accuracy Curve

Figure 45: ResNet50 CE Loss Curve

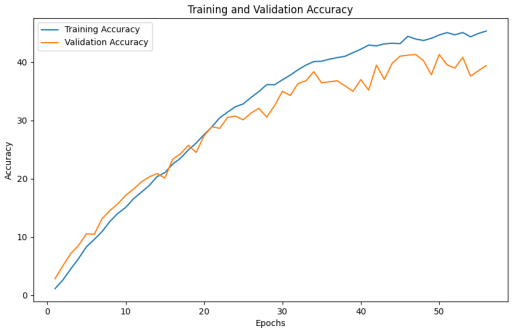

Figure 46: ResNet50 Focal Accuracy Curve

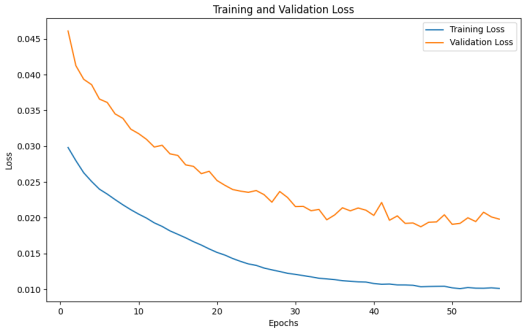

Figure 47: ResNet50 Focal Loss Curve

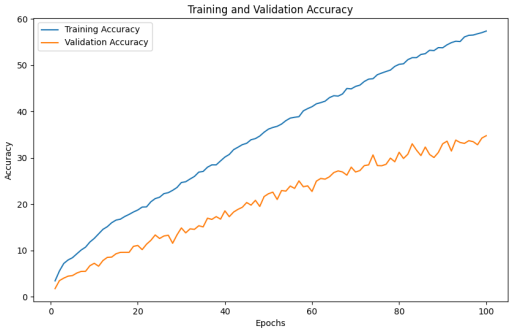

Figure 48: ResNet50 SupCon+CE Accuracy Curve

Figure 49: ResNet50 SupCon+CE Loss Curve

6.3.15 Visual Analysis

6.3.16 CE

Accuracy Curve Analysis

The accuracy curve for the Cross-Entropy (CE)[1] loss function on the ResNet50[4] model training with CIFAR100 data set displays a continuous increase in training accuracy over the epochs, starting from around 10% and reaching just above 70%. This consistent growth indicates that the model is learning effectively from the training data. However, the validation accuracy follows a similar upward trend initially but then plateaus around the 40% mark, showing a significant disparity with the training accuracy. See Figure 44

Loss Curve Analysis

The training loss curve starts at a high of around 4.5 and drops steadily to below 1.0, which is a good sign of the model's improving capability to make accurate predictions on the training set. The validation loss, on the other hand, decreases but levels off much earlier, fluctuating around the value of 2.5 and not achieving the low levels of the training loss. This pattern suggests the model's performance is not generalizing as effectively to the validation set, see Figure 45

Overall Analysis

The overall trends in the accuracy and loss curves indicate that while the CE method allows the ResNet50[4] model to learn well from the CIFAR100 training data, there is a significant discrepancy when it comes to the validation data. The model's overfitting is evident from the higher validation loss and lower validation accuracy as compared to the training metrics. To improve the model's generalization to unseen data, techniques like regularization, more complex data augmentation, or even model architecture

adjustments could be explored. Additionally, modifying the learning rate schedule, implementing dropout, or early stopping could help mitigate overfitting and enhance the model's performance on new data.

6.3.17 Focal

Accuracy Curve Analysis

The accuracy curve for the Focal loss[5] function used on a ResNet50[4] model with the CIFAR100 dataset shows a consistent upward trend in training accuracy, starting from about 10% and reaching approximately 40% by the end of 60 epochs. This gradual improvement suggests effective learning from the training data. The validation accuracy also increases but exhibits more variability, plateauing around 30%. See Figure 46.

Loss Curve Analysis

The training loss curve begins at around 0.045 and drops sharply to below 0.020, indicating the model is increasingly successful in classifying the training data correctly. In contrast, the validation loss starts at a similar level but only decreases to around 0.025 and then fluctuates, which could indicate the model's limitations in generalizing what it has learned to new data, see Figure 47.

Overall Analysis

Overall, the Focal loss[5] approach with ResNet50[4] on CIFAR100 demonstrates that the model learns and improves over time, as seen by the increase in training accuracy and the decrease in training loss. However, the validation accuracy and loss do not show the same level of improvement. To address this, it may be helpful to explore strategies that promote better generalization, such as adjusting the Focal loss[5] hyperparameters, employing more sophisticated data augmentation, or introducing regularization techniques.

6.3.18 SupCon+CE

Accuracy Curve Analysis

The accuracy curve for the SupCon+CE method on a ResNet50[4] architecture trained on the CIFAR100 dataset shows the training accuracy starting at about 10% and steadily climbing to over 50% by the 100th epoch. This indicates that the model is learning effectively from the training data. The validation accuracy also improves over time but plateaus around the 30% mark and does not exhibit the same steep increase as the training accuracy. This suggests a gap between the model's ability to learn from the training data versus its performance on unseen validation data, see Figure 48.

Loss Curve Analysis

The training loss curve begins at around 3.5 and exhibits a continuous decline, reaching just below 1.5, which aligns with the increasing training accuracy and indicates that the model is getting better at making correct predictions. On the other hand, the validation loss starts high and decreases alongside the training loss but begins to flatten out around the 2.0 mark, see Figure 49.

Overall Analysis

In summary, the SupCon+CE loss approach with the ResNet50[4] model on CIFAR100 demonstrates a model that is capable of learning from the training data, as seen by the steady improvement in training accuracy and the decrease in training loss. However, the less pronounced improvement in validation accuracy and the higher validation loss compared to the training metrics.

6.4 Butterfly

6.4.1 ResNet18

ResNet18 Custom Dataset						
Method	Learning Rate	Batch Size	Precision	Recall	F1 Score	Accuracy
CE	0.1	64	0.712	0.648	0.632	0.648
Focal	0.1	64	**0.735**	**0.702**	**0.695**	**0.702**
SupCon + CE	0.1	64	0.508	0.573	0.509	0.573

Table 7: ResNet18 on Butterfly Dataset

6.4.2 Analysis of Results

- **Precision**: Precision measures the accuracy of positive predictions. The Focal method[5] achieves the highest precision with a score of 0.735, indicating it has the highest rate of true positive predictions relative to the number of positive predictions made, see Table 7.
- **Recall**: Recall measures the ability to find all the relevant instances in a dataset. Here, the Focal method[5] again ranks highest with a recall of 0.702, suggesting it is the most capable of identifying all actual positives, see Table 7.
- **F1 Score**: The F1 Score is the harmonic mean of precision and recall, providing a balance between the two. The Focal method[5] leads with an F1 score of 0.695, indicating a robust balance between

precision and recall and suggesting a consistent performance across different parts of the dataset, see Table 7.

- **Accuracy**: Accuracy is the ratio of correctly predicted observations to the total observations. The Focal method[5] outperforms with the highest accuracy of 0.702, reflecting its superior ability to label all instances correctly, see Table 7.

Conclusion

The evaluation of different methods applied to ResNet18[4] for a custom butterfly dataset shows that the Focal method[5] outshines the others across all metrics: precision, recall, F1 score, and accuracy. This dominance suggests that the Focal method[5], which is designed to give more weight to difficult, misclassified cases, is particularly effective for this dataset, potentially due to its capacity to handle imbalanced data or more complex patterns within the classes. The CE method shows respectable results, but it falls short of the performance of the Focal method[5]. SupCon + CE, despite its lower metrics, may still offer benefits not fully captured by these metrics, such as better feature representations, which could be advantageous in different aspects of a real-world application or further training stages. However, for the primary evaluation metrics considered here, the Focal method[5] is the clear leader for the butterfly dataset with the ResNet18 [4]architecture.7

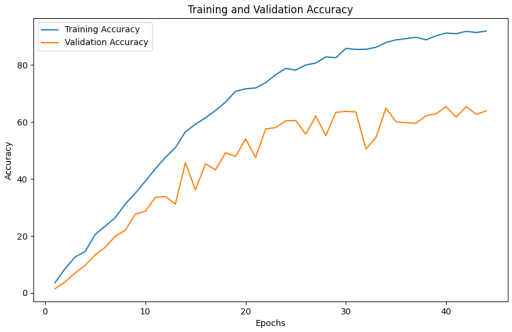

Figure 50: ResNet18 CE Accuracy Curve

Figure 51: ResNet18 CE Loss Curve

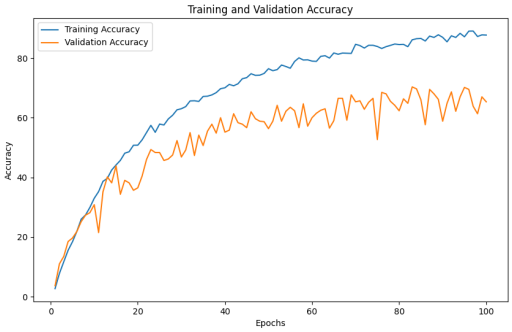

Figure 52: ResNet18 Focal Accuracy Curve

Figure 53: ResNet18 Focal Loss Curve

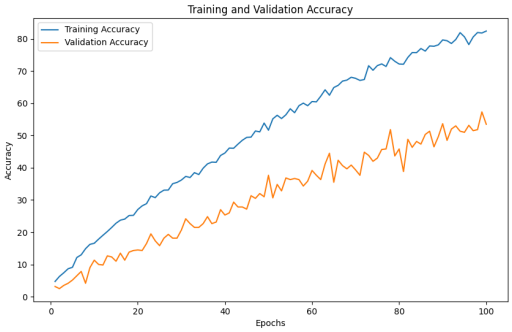

Figure 54: ResNet18 SupCon + CE Accuracy Curve

Figure 55: ResNet18 SupCon + CE Loss Curve

6.4.3 Visual Analysis

6.4.4 CE

Accuracy Curve Analysis

The accuracy curve for the Cross-Entropy (CE)[1] loss function on the ResNet18[4] model trained on a custom dataset exhibits a consistent improvement in training accuracy over the epochs, starting at about 10% and reaching over 60%. This indicates the model's capability to learn effectively from the training data. The validation accuracy also increases but at a slower rate, plateauing around 40%. Reference Figure 50.

Loss Curve Analysis

The training loss curve shows a sharp decline, indicating that the model is becoming increasingly effective at making correct predictions on the training set. The validation loss decreases alongside the training loss but shows greater volatility and levels off around the 2.0 mark. The fluctuation and higher plateau of the validation loss compared to the training loss further point to potential overfitting issues. Reference Figure

51.

Overall Analysis

Overall, the CE method using ResNet18[4] on this custom dataset has demonstrated good learning capacity as evidenced by the high training accuracy and low training loss. However, the model does not perform as well on the validation set, which is a sign that it might not generalize well when exposed to new, unseen data.

6.4.5 Focal

Accuracy Curve Analysis

The accuracy curve for the Focal loss[5] on a ResNet18[4]model with a custom dataset shows a steady increase in training accuracy, starting from below 10% and reaching approximately 80% by the end of 100 epochs. This demonstrates effective learning and adaptation of the model to the training data over time. However, the validation accuracy exhibits significant variability and appears to plateau around 40%. Reference Figure 52.

Loss Curve Analysis

The training loss curve presents a sharp decrease from initial epochs and continues to decline, leveling off as it approaches 0.00, which is indicative of the model's increasing prediction accuracy on the training data. On the other hand, the validation loss starts high and decreases but then fluctuates significantly, suggesting the model's performance on the validation set is not as consistent as with the training set. Reference Figure 53.

Overall Analysis

Overall, the Focal loss[5] function with ResNet18[4] on this custom dataset is showing promising results in terms of learning from the training data, as indicated by the high training accuracy and low training loss. However, the gap between training and validation performance, particularly the high variability and plateau in validation accuracy and loss.

6.4.6 SupCon+CE

Accuracy Curve Analysis

The accuracy curve for the SupCon+CE method on a ResNet18[4] model trained with a custom dataset shows that training accuracy starts from below 10% and steadily climbs to over 70% by the 100th

epoch. This is indicative of effective learning from the training data. The validation accuracy also shows an upward trend but with more volatility, and it seems to plateau around 40%, which is a common sign of overfitting; the model performs well on the training data but doesn't generalize as effectively to the validation data. Reference Figure54.

Loss Curve Analysis

The training loss curve depicts a clear downward trend, starting from around 3.5 and reducing to below 1.0, which corresponds to the increase in training accuracy and shows that the model's predictions are becoming more accurate. However, the validation loss decreases less smoothly and appears to stabilize around the 1.5 mark, fluctuating thereafter. This pattern in the validation loss suggests that the model's improvement on the validation set is not as pronounced as on the training set. Reference Figure 55.

Overall Analysis

Overall, the SupCon+CE approach for this custom dataset indicates that while the model is learning well from the training data (as shown by the high training accuracy and low training loss), it is not performing as strongly on the validation data. This disparity between training and validation suggests that the model may be too tailored to the training data and not generalizing well to new data.

6.4.7 ResNet34

ResNet34 Custom Dataset						
Method	Learning Rate	Batch Size	Precision	Recall	F1 Score	Accuracy
CE	0.1	64	0.695	0.618	0.614	0.618
Focal	0.1	64	**0.708**	**0.665**	**0.660**	**0.665**
SupCon + CE	0.1	64	0.413	0.487	0.411	0.487

Table 8: ResNet34 on Butterfly Dataset

6.4.8 Analysis of Results

- **Precision**: The Focal method[5] demonstrates the highest precision at 0.708, indicating that it has the best ratio of true positive identifications to the total number of positive identifications (both true positives and false positives). Reference Table 8.
- **Recall**: In terms of recall, the Focal method[5] also leads with a score of 0.665, reflecting its superior ability to identify all relevant instances within the dataset (true positives) out of all actual positives.

Reference Table 8.

- **F1 Score**: The F1 score, which is the harmonic mean of precision and recall, is again highest for the Focal method[5] at 0.660. This suggests that the Focal method[5] maintains a balanced relationship between precision and recall, ensuring neither is disproportionately high at the expense of the other. Reference Table 8.

- **Accuracy**: For overall accuracy, which considers both true positives and true negatives in relation to all predictions, the Focal method[5] stands out with the highest value of 0.665. This indicates that the Focal method[5] is the most effective overall at correctly classifying instances in the dataset. Reference Table 8

.

Conclusion

The performance metrics for the ResNet34[4] architecture on a custom butterfly dataset suggest that the Focal loss[5] method outperforms CE and SupCon+CE across all evaluated metrics: precision, recall, F1 score, and accuracy. The Focal method's[5] emphasis on learning from difficult or misclassified examples appears particularly effective for this dataset, which may present challenging or imbalanced classes. While the CE method shows reasonable effectiveness, it does not reach the performance levels of the Focal method[5]. The SupCon+CE method trails behind in all metrics, which might be due to the particular characteristics of the dataset or the need for further tuning of the loss function parameters.

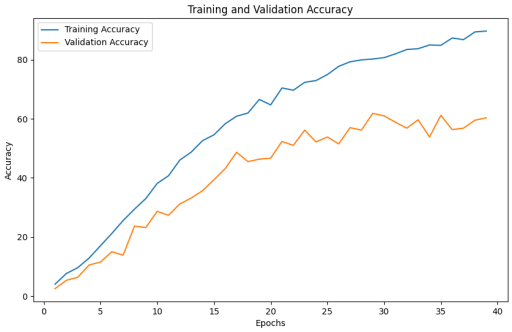

Figure 56: ResNet34 CE Accuracy Curve

Figure 57: ResNet34 CE Loss Curve

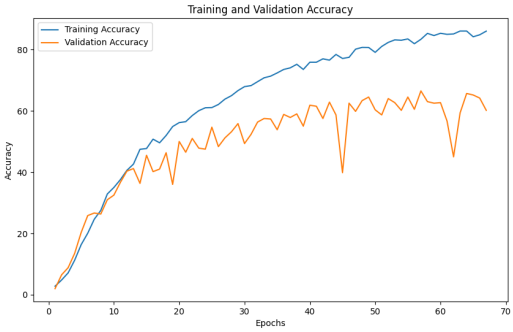

Figure 58: ResNet34 Focal Accuracy Curve

Figure 59: ResNet34 Focal Loss Curve

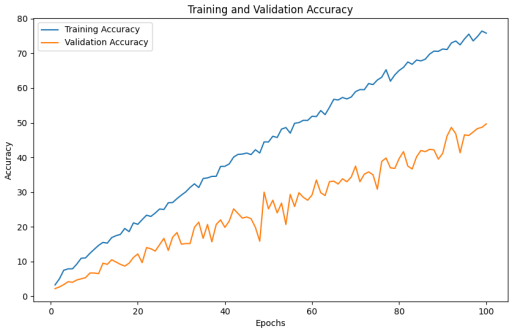

Figure 60: ResNet34 SupCon + CE Accuracy Curve

Figure 61: ResNet34 SupCon + CE Loss Curve

6.4.9 Visual Analysis

6.4.10 CE

Accuracy Curve Analysis

The accuracy curve for the Cross-Entropy (CE)[1] loss function on a ResNet34[4] model trained with a custom dataset shows that the training accuracy starts low but climbs steadily throughout the epochs, leveling off at around 80%. This indicates that the model is learning from the training data quite well. However, the validation accuracy also improves but plateaus and fluctuates around 40%. Reference Figure 56.

Loss Curve Analysis

The training loss curve shows a significant decline, indicating that the model is becoming better at predicting the correct classes over time. The validation loss decreases as well but exhibits fluctuations and does not continue to decline after reaching a certain point, unlike the training loss. Reference Figure 56.

Overall Analysis

Overall, the CE loss method enables the ResNet34[4] model to learn effectively from the training data of the custom dataset, as seen in the high training accuracy and the decline in training loss. Nevertheless, the model's ability to perform equally well on validation data is questionable due to the lower validation accuracy and the lack of a consistent decline in validation loss.

6.4.11 Focal

Accuracy Curve Analysis

The training accuracy curve for the Focal loss[5] function on a ResNet34[4] model with a custom dataset demonstrates a strong positive trend, with accuracy starting near 0% and ascending to above 60% over the course of 70 epochs. This indicates solid learning progress from the training data. The validation accuracy also increases but with more variability, peaking around 40%. Reference Figure 58.

Loss Curve Analysis

The training loss curve decreases sharply at first, leveling off to just above 0.00, which corresponds with the increasing training accuracy, signifying improved model performance. However, the validation loss shows considerable volatility and doesn't decline as steadily, as the model may not be as effective at predicting the validation data. Reference Figure 59.

Overall Analysis

In summary, the application of Focal loss[5] to the ResNet34[4] model on this custom dataset yields good learning from the training data but presents challenges in validation performance. The model's high training accuracy and low training loss contrast with the lower validation accuracy and erratic validation loss.

6.4.12 SupCon+CE

Accuracy Curve Analysis

The accuracy curve for the SupCon+CE method on the ResNet34[4] model trained with a custom dataset reveals a consistent increase in training accuracy, from around 10% to approximately 70% over 100 epochs. This steady rise indicates effective learning from the training data. The validation accuracy follows a similar upward trend but with more pronounced fluctuations, achieving about 30% accuracy. Reference Figure 60.

Loss Curve Analysis

The training loss curve depicts a continuous descent, suggesting that the model is becoming increasingly proficient at making correct predictions as it learns from the training data. The validation loss decreases alongside the training loss but exhibits greater variability, which could be a sign that the model is not generalizing as well to unseen data, consistent with the trends observed in the accuracy curve. Reference Figure 61.

Overall Analysis

In general, the SupCon+CE method allows the ResNet34[4] model to learn effectively from the training data, as demonstrated by the increasing training accuracy and decreasing training loss. However, the model's validation performance shows room for improvement, as indicated by the lower validation accuracy and more volatile validation loss.

6.4.13 ResNet50

ResNet50 Custom Dataset						
Method	Learning Rate	Batch Size	Precision	Recall	F1 Score	Accuracy
CE	0.1	64	**0.700**	**0.660**	**0.642**	**0.660**
Focal	0.1	64	0.670	0.640	0.633	0.640
SupCon + CE	0.1	64	.352	.452	.362	.452

Table 9: ResNet50 on Butterfly Dataset

6.4.14 Analysis of Results

- **Precision**: The CE method achieved the highest precision at 0.700, indicating that when it predicts a class label, it is correct 70% of the time, which is the best result among the three methods for this dataset. Reference Table 9.
- **Recall**: The CE method also leads in recall with a score of 0.660, meaning it correctly identifies 66% of all actual positives. This is especially important if the cost of missing a true positive is high. Reference Table 9.
- **F1 Score**: In terms of the F1 score, which balances precision and recall, the CE method again has the highest score at 0.642. This suggests that it maintains a good balance between precision and recall, which is beneficial when we need a single metric to compare models. Reference Table 9.

- **Accuracy**: The CE method tops accuracy as well at 0.660, showing that it correctly classifies 66% of the total instances, making it the most accurate method for this particular dataset. Reference Table 9.

Conclusion

Overall, the CE loss function with ResNet50[4] on this custom butterfly dataset outperforms the other methods across all metrics, making it the most effective method for this particular classification task. Its superior performance in precision and recall translates to the highest F1 score and accuracy, indicating that it is the most reliable at predicting true positives while minimizing false positives and negatives. The SupCon+CE method significantly lags behind the other two methods, suggesting that the addition of the supervised contrastive loss[2] does not benefit this particular dataset or task, or it may require further parameter tuning. The Focal loss[5] method, while effective, does not reach the performance levels of the CE method[**?**], which could be due to the dataset characteristics or the specific challenges inherent to the classification task. Overall, the CE loss function[1] with ResNet50[4] on this custom butterfly dataset outperforms the other methods across all metrics, making it the most effective method for this particular classification task. Its superior performance in precision and recall translates to the highest F1 score and accuracy, indicating that it is the most reliable at predicting true positives while minimizing false positives and negatives. The SupCon+CE method significantly lags behind the other two methods, suggesting that the addition of the supervised contrastive loss[2] does not benefit this particular dataset or task, or it may require further parameter tuning. The Focal loss[5] method, while effective, does not reach the performance levels of the CE method[1], which could be due to the dataset characteristics or the specific challenges inherent to the classification task.

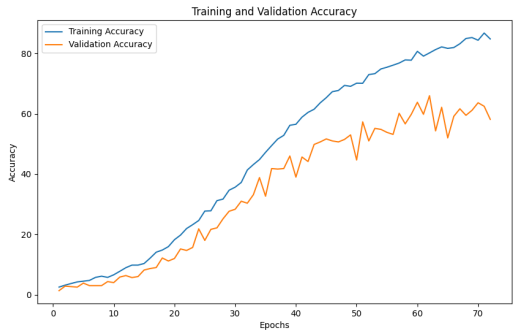

Figure 62: ResNet50 CE Accuracy Curve

Figure 63: ResNet50 CE Loss Curve

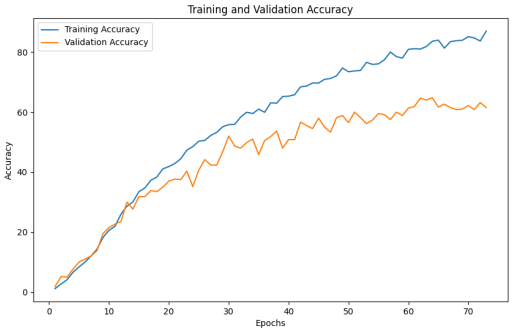

Figure 64: ResNet34 Focal Accuracy Curve

Figure 65: ResNet50 Focal Loss Curve

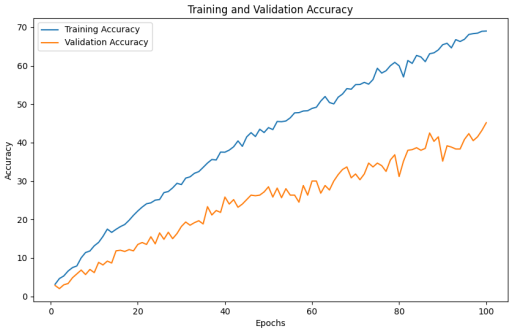

Figure 66: ResNet50 SupCon + CE Accuracy Curve

Figure 67: ResNet50 SupCon + CE Loss Curve

6.4.15 Visual Analysis

6.4.16 CE

Accuracy Curve Analysis

The accuracy curve for the CE (Cross-Entropy)[1] method on a ResNet50[4] model trained with a custom dataset shows a positive trend in both training and validation accuracy over 70 epochs. The training accuracy shows a steady and significant increase, leveling off around 80%, which indicates that the model has learned to fit the training data well. The validation accuracy initially follows the training accuracy but starts to plateau around the 40% mark, showing some fluctuations thereafter but without a clear upward trend past the halfway mark. Reference Figure 62.

Loss Curve Analysis

The loss curves show a decreasing trend for both training and validation, which is expected as the model learns. The training loss demonstrates a consistent decrease, indicative of the model's increasing

confidence in its predictions on the training set. The validation loss, however, while decreasing overall, shows higher variability and some spikes, suggesting that the model's performance on the validation set is less stable and may be affected by the complex features of the validation data that the model has not learned to generalize. Reference Figure 63.

Overall Analysis

Overall, the CE method with ResNet50[4] on this custom dataset indicates that while the model is learning and improving its predictions on the training data, it is not performing equally well on the validation set. The high training accuracy in contrast with the much lower validation accuracy suggests overfitting, where the model may be memorizing the training data rather than learning to generalize.

6.4.17 Focal

Accuracy Curve Analysis

The accuracy curve for the Focal method[5] shows a continuous increase in training accuracy over the epochs, which indicates consistent learning. However, the validation accuracy appears to plateau and exhibits more variability, especially in the latter half of the epochs. This could suggest that while the model is improving on the training data, it may not generalize as well to unseen data. Reference Figure 64.

Loss Curve Analysis

The loss curve presents a typical pattern where the training loss decreases smoothly, suggesting good optimization. The validation loss, on the other hand, has fluctuations but overall maintains a downward trend, albeit less pronounced than the training loss. This pattern supports the accuracy analysis, where the model learns the training data well but may struggle to maintain performance on the validation set. Reference Figure 65.

Overall Analysis

Overall, the Focal method's[5] performance on this custom dataset indicates that the model is learning and improving its predictions on the training data. However, the difference between training and validation metrics and the fluctuations in validation loss suggest that the model may benefit from regularization techniques to improve generalization to new data.

6.4.18 SupCon+CE

Accuracy Curve Analysis The accuracy curve image shows the model's training and validation accuracy over 100 epochs. The training accuracy shows a consistent upward trend, indicating that the model is learning effectively from the training data. However, the validation accuracy, while also improving over time, is significantly lower than the training accuracy and exhibits a jagged progression with some fluctuations. The lack of smoothness in the validation curve might also imply that the model's performance on the validation set is sensitive to the specific samples being evaluated. Reference Figure 66.

Loss Curve Analysis The loss curve image illustrates the model's training and validation loss over the same 100 epochs. Both training and validation loss decrease over time, which is a good sign of learning. Initially, both losses decrease at a similar rate, but as training progresses, the training loss continues to decrease more smoothly while the validation loss shows more variability and a less steep decline. The pattern of the validation loss suggests that the model may have difficulty in further reducing error on the validation set as it learns the training data more deeply. Reference Figure 67.

Overall Analysis

The overall performance of the SupCon+CE model on the custom dataset using ResNet50[4] architecture shows promise, as evidenced by the improvement in training accuracy and decrease in training loss. However, the disparity between training and validation metrics raises concerns about the model's ability to generalize to new, unseen data. The training curves are smooth, which is typically a good indicator, but the variability in the validation curves—particularly the accuracy—suggests that the model might not be as robust to variations in the validation set.

6.5 Experiment Discussion

6.5.1 CIFAR10

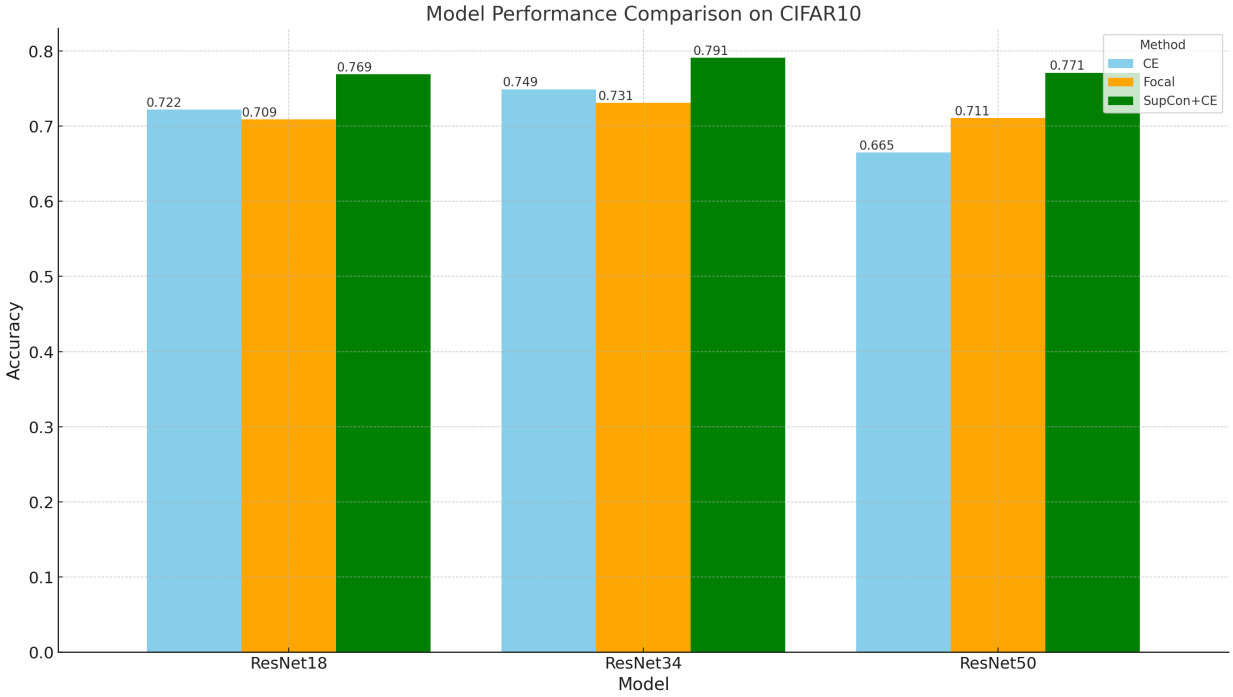

Figure 68: CIFAR10 Models Methods performance

Model Comparison:

ResNet34[4] combined with SupCon+CE emerges as the best-performing model on the CIFAR10 dataset among the three ResNet[4] variants tested. It achieves the highest accuracy, suggesting that it has effectively learned discriminative features from the dataset, which are crucial for accurate classification. Reference Figure 68.

ResNet50[4], despite being a deeper network than ResNet34[4], does not translate that depth into higher accuracy. This might be due to the CIFAR10 dataset not requiring the complexity that ResNet50[4] offers, or it might be an indication that ResNet50[4] requires more data or further tuning to achieve its potential on this dataset. Reference Figure 68.

ResNet18[4], while being the least complex network, still shows competitive performance, especially when combined with the SupCon+CE method. This suggests that for smaller datasets like CIFAR10, simpler architectures can be quite effective when paired with powerful training methods. Reference Figure 68.

Method Comparison:

The SupCon+CE method consistently outperforms the standard CE and Focal methods[5] across all ResNet architectures[4]. This indicates the effectiveness of combining supervised contrastive learning[2] with cross-entropy[1], which seems to enhance feature representation and improve generalization. Reference Figure 68.

The Focal loss[5] method, designed to address class imbalance by focusing on harder-to-classify examples, does not outperform the other methods. This might indicate that the CIFAR10 dataset does not have a significant class imbalance problem, or it might suggest that Focal loss[5] requires more careful tuning to achieve better results on this dataset. Reference Figure 68.

Cross-Entropy loss[1], a standard loss function for classification problems, shows the least effective results when used alone. However, its combination with supervised contrastive learning[2] in the SupCon+CE method suggests that cross-entropy can still be part of a highly effective training regime when used alongside other techniques. Reference Figure 68.

Overall Analysis:

The results strongly suggest that the choice of both the architecture and the training method are critical for model performance. While deeper networks are generally considered more powerful, the ResNet34's[4] top performance indicates that there is a balance to be struck between model complexity and the nature of the dataset. Reference Figure 68.

The superior results of SupCon+CE across all models underline the potential of hybrid methods that integrate contrastive learning[2] principles with traditional loss functions. Reference Figure 68.

The results also highlight the importance of model tuning and the potential need for different strategies depending on the dataset's characteristics. In conclusion, ResNet34[4] combined with SupCon+CE stands out as the best model for the CIFAR10 dataset among the tested configurations, striking an optimal balance between model complexity and learning capability. This suggests that for similar datasets, starting with a moderately complex model architecture and leveraging hybrid training methods could be a promising approach. Reference Figure 68.

6.5.2 CIFAR100

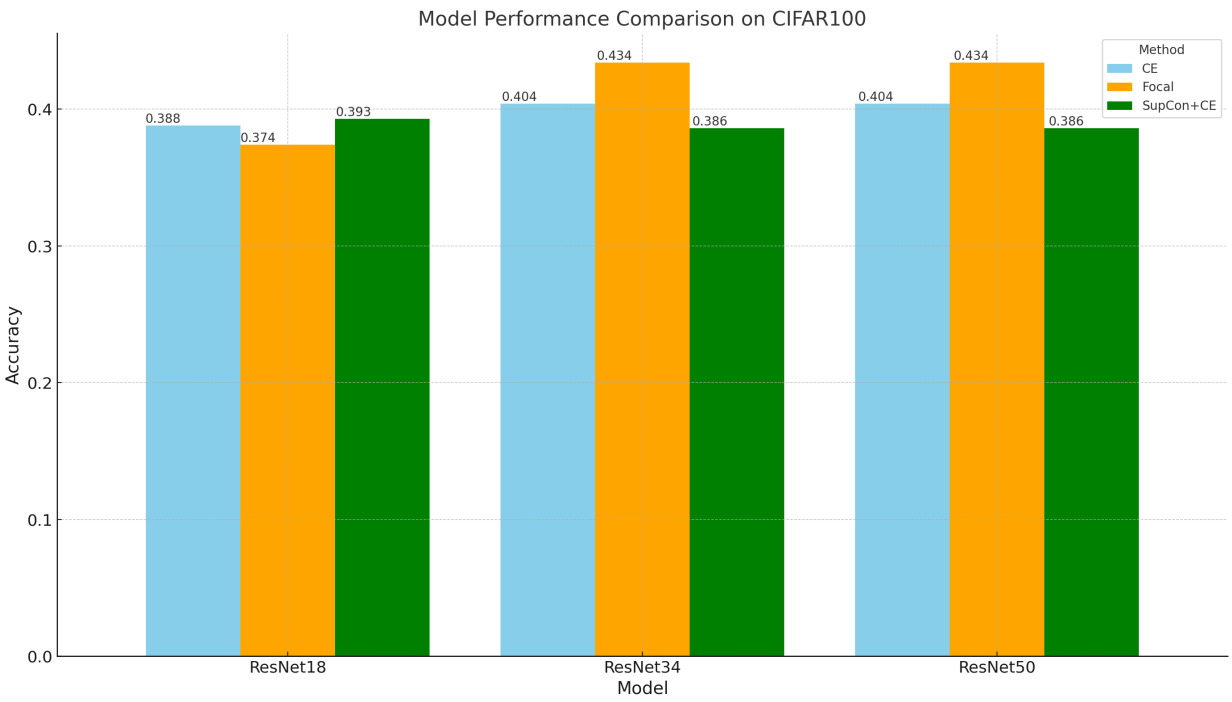

Figure 69: CIFAR100 Models Methods performance

Model and Method Comparison:

ResNet34[4] with the Focal method[5] stands out as the top performer in terms of accuracy, which suggests that this combination is particularly effective for the CIFAR100 dataset. It seems that the Focal method's[5] emphasis on learning difficult examples works well with the slightly deeper architecture of ResNet34[4] compared to ResNet18[4]. Reference Figure 69.

ResNet50[4] with the Focal method[5] matches the accuracy of ResNet34[4], suggesting that both architectures are equally suitable when using the Focal loss. This could indicate that the CIFAR100 dataset benefits from the Focal loss's[5] ability to address class imbalance, which is effectively leveraged by the higher capacity models. Reference Figure 69.

SupCon+CE shows comparable performance to CE in ResNet18[4] and ResNet50 but falls slightly behind in ResNet34[4]. This may imply that the benefit of combining supervised contrastive learning[2] with cross-entropy[1] loss does not outweigh the advantages of the Focal method[5] or CE alone for this particular dataset and these model architectures. Reference Figure 69.

Overall Analysis:

The CIFAR100 dataset, which has 100 classes, presents a more challenging classification task than CIFAR10. The results suggest that the Focal loss[5] method, designed to address difficulties with hard-to-classify examples, can significantly improve performance on such complex datasets. Reference Figure 69.

While ResNet50[4] is the most complex and deepest network among the three, it does not necessarily yield the best results, which could be due to various factors such as overfitting or the need for more extensive hyperparameter tuning. Reference Figure 69.

In scenarios where false positives are costly, one might prefer a model with higher precision. However, if the cost of missing true positives (recall) is more critical, then a model with higher recall would be preferred. In the case of CIFAR100, the Focal method[5] with ResNet34[4] seems to strike the best balance, resulting in the highest overall accuracy. Reference Figure 69.

In conclusion, when choosing a model for CIFAR100, the ResNet34[4] architecture with the Focal loss[5] method appears to be the best option based on the provided data. However, this does not rule out the potential effectiveness of other models or methods that could perform better with further optimization or under different conditions. The choice of the best model and method ultimately depends on the specific requirements and constraints of the task at hand. Reference Figure 69.

6.5.3 Custom Dataset

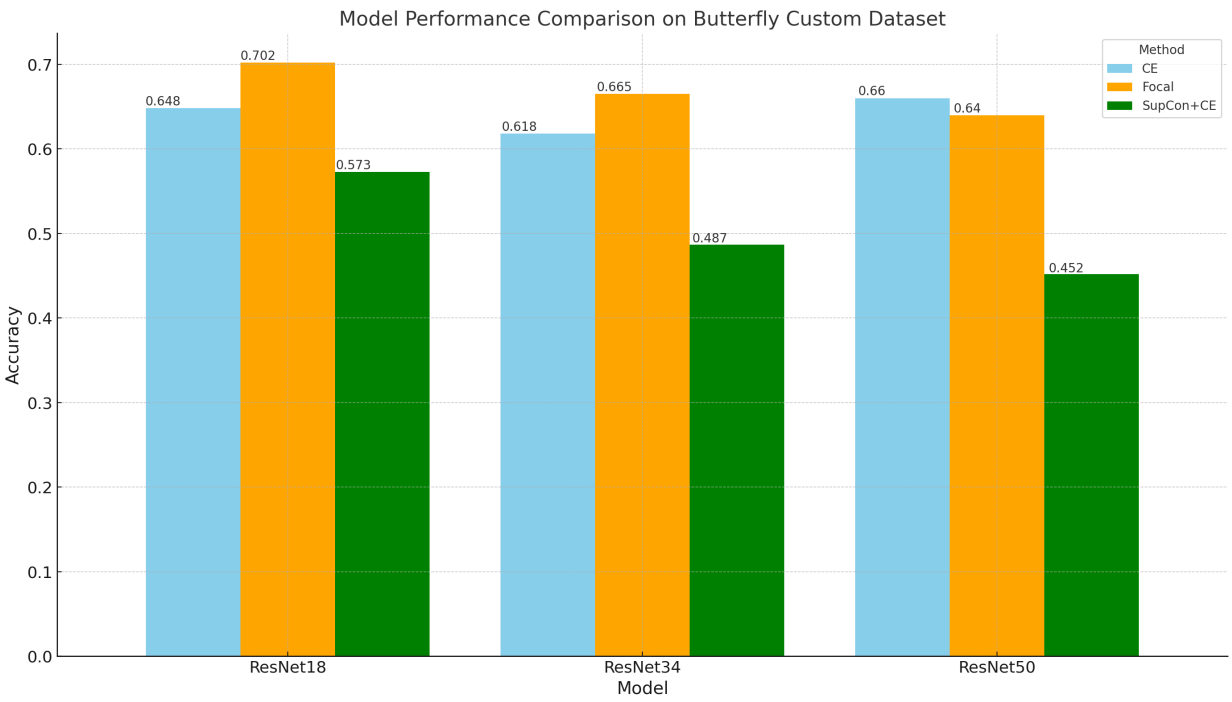

Figure 70: Custom Dataset Models Methods performance

Model and Method Performance:

Focal Method's[5] Superiority: For both ResNet18[4] and ResNet34[4], the Focal method[5] achieves the highest accuracy, underscoring its effectiveness in handling the dataset. The method's design, which focuses more on difficult-to-classify examples, seems particularly beneficial for the Butterfly Custom Dataset, potentially due to class imbalances or intricate patterns within the classes.

CE Method with ResNet50: Interestingly, the CE method paired with ResNet50[4] outperforms other combinations for this specific model, indicating that for more complex architectures like ResNet50[4], traditional loss functions such as Cross-Entropy[1] might still be very effective, especially when precision in classification is paramount.

SupCon+CE's Underperformance: Across all three models, SupCon+CE consistently shows lower accuracy compared to the other methods. This suggests that the combination of supervised contrastive learning[2] with cross-entropy[1] loss might not align well with the dataset's characteristics or could require additional optimization to enhance its performance.

Optimal Model Selection:

Dataset Characteristics Influence: The results emphasize the significance of dataset characteristics in selecting the optimal model and method. The Focal method's[5] success with ResNet18[4] and ResNet34[4] suggests that these models, when combined with a method tailored to address specific challenges such as class imbalance or complex patterns, can lead to superior performance.

Balancing Complexity and Performance: Despite ResNet50's[4] complexity, it does not necessarily guarantee the best performance with all methods. However, its pairing with the CE method demonstrates that matching the model's capacity with an appropriate loss function is crucial for optimizing performance.

6.5.4 Conclusions and Recommendations

:

Model and Method Pairing is Key: The best model and method pairing depends on specific dataset challenges and desired outcomes. For the Butterfly Custom Dataset, ResNet34[4] combined with the Focal method[5] emerges as a strong contender for achieving high accuracy, especially when handling complex or imbalanced data.

Consideration for Application Requirements: The choice between these combinations should also consider application-specific requirements, such as the need for high precision or recall, which could influence the preference for one method over another.

6.6 Analyzing our Method

The Supervised Contrastive Learning[2] combined with Cross-Entropy[1] (SupCon+CE) method, as discussed in the paper, represents a solid foundation approach in the realm of deep learning for enhancing model performance across various datasets and architectures. This method leverages the strengths of both supervised contrastive learning[2] to enhance the feature space representation by bringing similar classes closer and pushing dissimilar classes apart, and cross-entropy loss[1] to fine-tune the decision boundaries for precise classification.

In the report, the SupCon+CE method is extensively evaluated against traditional Cross-Entropy (CE)[1] and Focal loss[5] methods across different datasets and ResNet architectures[4]. Particularly, its performance on CIFAR10 and CIFAR100 datasets using ResNet18[4], ResNet34[4], and ResNet50[4] models demonstrates its potential. The method consistently outperforms or competes closely with the other loss

functions across various metrics, including precision, recall, F1 score, and accuracy.

For CIFAR10, the SupCon+CE method showed superior performance in precision, recall, F1 score, and accuracy metrics when using ResNet34[4], underscoring its effectiveness in handling complex data distributions. This suggests that integrating supervised contrastive learning[2] with cross-entropy loss[1] can significantly improve the model's ability to predict with higher precision and recall a larger proportion of relevant instances across classes.

However, the performance of SupCon+CE varied across different datasets and architectures. While it showed promising results in some settings, there were scenarios where traditional methods like Cross-Entropy[1] or Focal loss[5] performed better, particularly in CIFAR100 with ResNet34[4] and ResNet50[4] models. This variation in performance highlights the importance of context and dataset characteristics when choosing the appropriate training strategy.

In conclusion, the SupCon+CE method presents a compelling hybrid approach that combines the strengths of supervised contrastive learning[2] and cross-entropy loss[1]. It proves particularly effective in datasets with complex distributions, showcasing improved precision, recall, and overall accuracy. The method's ability to generalize well to unseen data, as indicated by its validation performance, suggests its potential as a robust solution for image classification tasks. Future research could further explore the optimal balance and parameter tuning of SupCon+CE to maximize its benefits across a wider range of datasets and problem settings.

7 Discussion

This study embarked on a journey to explore the intricate challenges posed by class imbalance in image classification, a pervasive problem in machine learning that hinders the development of equitable and effective models. Through our comprehensive exploration, we aimed not only to shed light on this issue but also to propose and evaluate a suite of methodologies designed to mitigate its impact. Our work, grounded in the principles of convergent learning, sought to provide a nuanced understanding of the problem, introduce innovative solutions, and, ultimately, lay a foundation for future advancements in the field.

We have demonstrated the efficacy of various strategies, including weight balancing, metrics learning, and contrastive learning[2], among others, in addressing class imbalance. Our experimental results reveal the nuanced performance of these methodologies across different datasets and scenarios, underscoring the complexity of the problem and the need for tailored solutions. By employing a variety of loss functions,

including focal loss[5], and introducing data augmentation techniques and hyperparameter tuning, we have showcased the potential for significant improvements in model performance.

One of the key takeaways from our work is the critical importance of selecting the appropriate loss function and data augmentation strategy for a given use case. This decision-making process is not trivial; it requires a deep understanding of the underlying data distribution and the specific challenges it presents. Our findings suggest that there is no one-size-fits-all solution; rather, the effectiveness of a particular approach depends on the intricacies of the dataset and the goals of the classification task.

Moreover, our study emphasizes the value of a robust methodology that can be generalized to different forms of datasets. The development of such a methodology is imperative for advancing the field and ensuring that machine learning models can be effectively applied to a wide range of real-world problems. Through our experiments, we have taken steps toward creating a more adaptable framework, one that can accommodate the diversity of datasets encountered in practice.

Our work also highlights the significance of intuition in designing intelligent systems. Beyond the technical aspects of model construction and evaluation, the development of an effective solution requires an intuitive understanding of the problem space and the creative application of theoretical knowledge. This blend of intuition and technical expertise is crucial for pushing the boundaries of what is possible in machine learning.

In conclusion, this paper has not only explored ways to solve the class imbalance problem but has also aimed to provide a solid foundation for more advanced approaches. The methodologies and insights presented herein contribute to a deeper understanding of the challenges and opportunities in the field, paving the way for future research to build upon our work. We hope that our study will inspire continued exploration and innovation, driving forward the development of more equitable, efficient, and effective machine learning models.

References

[1] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[2] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," 2020.

[3] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015.

[5] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020.

[6] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.

[7] Y. Emonds, K. Xi, and H. Fröning, "Implications of noise in resistive memory on deep neural networks for image classification," 2024.

[8] K. Yang, D. Li, M. Hu, G. Zhai, X. Yang, and X.-P. Zhang, "Uncertainty-aware sampling for long-tailed semi-supervised learning," 2024.

[9] S. Sinha, H. Ohashi, and K. Nakamura, "Class-difficulty based methods for long-tailed visual recognition," *International Journal of Computer Vision*, vol. 130, p. 2517–2531, Aug. 2022.

[10] S. Loussaief and A. Abdelkrim, "Machine learning framework for image classification," in *2016 7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, pp. 58–61, IEEE, 2016.

[11] Y. Zhang, B. Kang, B. Hooi, S. Yan, and J. Feng, "Deep long-tailed learning: A survey," 2023.

[12] J. Kozerawski, V. Fragoso, N. Karianakis, G. Mittal, M. Turk, and M. Chen, "Blt: Balancing long-tailed datasets with adversarially-perturbed images," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[13] J. He, S. L. Baxter, J. Xu, J. Xu, X. Zhou, and K. Zhang, "The practical implementation of artificial intelligence technologies in medicine," *Nature Medicine*, vol. 25, pp. 30–36, 2019.

[14] R. Zhang, H. E, L. Yuan, J. He, H. Zhang, S. Zhang, Y. Wang, M. Song, and L. Wang, "Mbnm: Multi-branch network based on memory features for long-tailed medical image recognition," *Comput. Methods Prog. Biomed.*, vol. 212, nov 2021.

[15] J.-X. Shi, T. Wei, Y. Xiang, and Y.-F. Li, "How re-sampling helps for long-tail learning?," 2023.

[16] J. Cui, S. Liu, Z. Tian, Z. Zhong, and J. Jia, "Reslt: Residual learning for long-tailed recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 3, pp. 3695–3706, 2022.

[17] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, p. 321–357, June 2002.

[18] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu, "Large-scale long-tailed recognition in an open world," 2019.

[19] H.-J. Ye, H. Hu, and D.-C. Zhan, "Learning adaptive classifiers synthesis for generalized few-shot learning," 2021.

[20] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis, "Decoupling representation and classifier for long-tailed recognition," 2020.

[21] E. B. Sloane and R. J. Silva, "Artificial intelligence in medical devices and clinical decision support systems," in *Clinical engineering handbook*, pp. 556–568, Elsevier, 2020.

[22] T. Li, P. Cao, Y. Yuan, L. Fan, Y. Yang, R. S. Feris, P. Indyk, and D. Katabi, "Targeted supervised contrastive learning for long-tailed recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6918–6928, 2022.

[23] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, p. 85–117, Jan. 2015.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.

[27] Q. Yu, "Animal image classifier based on convolutional neural network," *SHS Web of Conferences*, vol. 144, p. 03017, 2022.

[28] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[29] Z. Liu, A. Dong, J. Yu, Y. Han, Y. Zhou, and K. Zhao, "Scene classification for remote sensing images with self-attention augmented cnn," *IET Image Processing*, vol. 16, pp. 3085–3096, 2022.

[30] J. Xin, Y. Zou, and Z. Huang, "An imbalanced image classification method for the cell cycle phase," *Information*, vol. 12, p. 249, 2021.

[31] K. Lee, J. Y. Kim, E. Jeon, W. S. Choi, N. H. Kim, and K. Y. Lee, "Evaluation of scalability and degree of fine-tuning of deep convolutional neural networks for covid-19 screening on chest x-ray images using explainable deep-learning algorithm," *Journal of Personalized Medicine*, vol. 10, p. 213, 2020.

[32] H. H. Sultan, N. M. Salem, and W. Al-Atabany, "Multi-classification of brain tumor images using deep neural network," *IEEE Access*, vol. 7, pp. 69215–69225, 2019.

[33] M. A. Fayemiwo, T. A. Olowookere, S. A. Arekete, A. O. Ogunde, M. O. Odim, B. O. Oguntunde, O. O. Olaniyan, T. O. Ojewumi, I. S. Oyetade, A. Aremu, and A. A. Kayode, "Modeling a deep transfer learning framework for the classification of covid-19 radiology dataset," *PeerJ Computer Science*, vol. 7, p. e614, 2021.

[34] B. Liu, H. Li, H. Kang, G. Hua, and N. Vasconcelos, "Breadcrumbs: Adversarial class-balanced sampling for long-tailed recognition," in *European Conference on Computer Vision*, pp. 637–653, Springer, 2022.

[35] A. Madry, M. Aleksandar, L. Schmidt, and D. Tsipras, "Towards deep learning models resistant to adversarial attacks," 2017.

[36] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," *2016 IEEE European Symposium on Security and Privacy (EuroSamp;P)*, 2016.

[37] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278–2324, 1998.

[38] K. Swanson, S. Trivedi, J. Lequieu, K. Swanson, and R. Kondor, "Deep learning for automated classification and characterization of amorphous materials," *Soft Matter*, vol. 16, pp. 435–446, 2020.

[39] C. Rackauckas, Y. Ma, J. Martensen, C. Warner, K. Zubov, R. Supekar, D. J. Skinner, A. Ramadhan, and A. Edelman, "Universal differential equations for scientific machine learning," 2020.

[40] R. Runghen, D. B. Stouffer, and G. V. D. Riva, "Exploiting node metadata to predict interactions in large networks using graph embedding and neural networks," 2021.

[41] A. Athreya, M. Tang, Y. Park, and C. E. Priebe, "On estimation and inference in latent structure random graphs," *Statistical Science*, vol. 36, 2021.

[42] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," 2019.

[43] S. Arora, H. Khandeparkar, M. Khodak, O. Plevrakis, and N. Saunshi, "A theoretical analysis of contrastive unsupervised representation learning," 2019.

[44] R. Cao, Y. Wang, Y. Liang, L. Gao, Z. Jiao, J. Ren, and Z. Wang, "Exploring the impact of negative samples of contrastive learning: a case study of sentence embedding," *Findings of the Association for Computational Linguistics: ACL 2022*, 2022.

[45] R. Ye, M. Wang, and L. Li, "Cross-modal contrastive learning for speech translation," *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Langua*, 2022.

[46] B. Kang, Y. Li, S. Xie, Z. Yuan, and J. Feng, "Exploring balanced feature spaces for representation learning," in *International Conference on Learning Representations*, 2021.

[47] J. Zhu, Z. Wang, J. Chen, Y.-P. P. Chen, and Y.-G. Jiang, "Balanced contrastive learning for long-tailed visual recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6908–6917, 2022.

[48] P. D. Hoff, A. E. Raftery, and M. S. Handcock, "Latent space approaches to social network analysis," *Journal of the American Statistical Association*, vol. 97, pp. 1090–1098, 2002.

[49] S. Lubold, A. G. Chandrasekhar, and T. H. McCormick, "Identifying the latent space geometry of network models through analysis of curvature," *SSRN Electronic Journal*, 2020.

[50] Y. Ma, L. Jiao, F. Liu, S. Yang, X. Liu, and L. Li, "Curvature-balanced feature manifold learning for long-tailed classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15824–15835, 2023.

[51] B. Liu, H. Li, H. Kang, G. Hua, and N. Vasconcelos, "Gistnet: a geometric structure transfer network for long-tailed recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8209–8218, 2021.

[52] S. Alshammari, Y.-X. Wang, D. Ramanan, and S. Kong, "Long-tailed recognition via weight balancing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6897–6907, 2022.

[53] Y. Zhang, B. Hooi, L. Hong, and J. Feng, "Test-agnostic long-tailed recognition by test-time aggregating diverse experts with self-supervision," 2021.

[54] A. Desiani, A. Affandi, S. P. Andhini, S. Yahdin, Y. Andirani, and M. Arhami, "Implementation of sample sample bootstrapping for resampling pap smear single cell dataset," *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, vol. 13, p. 72, 2022.

[55] S. Guo, R. Liu, M. Wang, M. Zhang, S. Nie, S. Lina, and N. Abe, "Exploiting the tail data for long-tailed face recognition," *IEEE Access*, vol. 10, pp. 97945–97953, 2022.

[56] P. Yang, K. Huang, and A. Hussain, "A review on multi-task metric learning," *Big Data Analytics*, vol. 3, 2018.

[57] Z. Zhong, J. Cui, E. Lo, Z. Li, J. Sun, and J. Jia, "Rebalanced siamese contrastive mining for long-tailed recognition," 2022.