Long-Tail Image Classification: Comparative Analysis of Data Augmentation, Generation, and Contrastive Learning Approaches

Scott Pozder
Northeastern University
Boston, MA 02115
pozder.sc@northeastern.edu

Kevin Tang
Northeastern University
Boston, MA 02115
tang.kevi@northeastern.edu

Abstract

Long-tail image classification suffers from severe class imbalance, hindering performance on underrepresented categories. In this work, we address this challenge by combining class-specific diffusion models and contrastive self-supervised learning. Our diffusion pipeline generates high-fidelity synthetic samples to balance the data distribution, while DINO enables robust feature learning without labels. Experiments on CIFAR-100 show notable improvements in overall and tail-class accuracy, with diffusion-based augmentation outperforming traditional methods. Although limited by hardware constraints, our findings highlight the potential of generative and contrastive techniques for scalable, real-world long-tail classification. Future work includes multi-class diffusion models and hybrid augmentation for broader applicability.

1 Introduction

The challenge of long-tail image classification, where class distributions are heavily imbalanced, remains a persistent obstacle in computer vision systems. This imbalance, common in real-world datasets, significantly impacts model performance on underrepresented classes, creating a critical gap between study benchmarks and practical applications. Our research addresses this challenge through a comparative analysis of multiple approaches using a modified version of the CIFAR-100 dataset [1] to exhibit a long-tail distribution ranging from 500 to 25 images per class; see Figure 1.

Long-tail distributions are prevalent across domains including autonomous driving, where rare objects like emergency vehicles must be reliably detected despite limited training examples; robotics systems that must recognize uncommon objects in varied environments; and medical imaging, where rare conditions may have limited available samples. Despite their importance, models trained on such imbalanced datasets typically underperform on minority classes, reinforcing the need for specialized techniques to address this disparity. Ideally, machine learning systems would be able to replicate the human's ability to recognize objects by matching perceived object properties to existing representations in memory to achieve one-shot object classification generalization.

This paper evaluates three distinct approaches to mitigate the challenges of long-tail classification: traditional data augmentation techniques (including flipping and cropping) [2], synthetic data generation using a U-Net diffusion model [3], and contrastive learning through DINO architecture [4]. By implementing these methods with a ResNet-50 backbone [5], we establish a comprehensive framework for analyzing their relative effectiveness and identifying the contexts in which each approach excels.

Our contributions are threefold:

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

1. We present a systematic comparison of augmentation, generation, and contrastive learning approaches on a controlled long-tail variant of CIFAR-100, providing insights into their relative strengths and limitations.
2. We demonstrate that synthetic data generation via diffusion models yields superior performance on minority classes compared to traditional augmentation techniques alone, suggesting a promising direction for addressing class imbalance.
3. We identify and analyze failure cases across methods, providing insights into the fundamental limitations of each approach and suggesting promising directions for future research in long-tail classification.

This work builds upon recent advances in image classification [c], data augmentation [c], generative models [c], and contrastive learning [c], while specifically addressing the challenge of long-tail distributions in practical vision systems. Our findings suggest that synthetic data generation, when properly implemented, provides a robust solution to the long-tail problem while maintaining computational feasibility.

Disclosure: This project is based on open-source implementations, including ResNet, U-Net image diffusion, DINO, and PyTorch libraries.

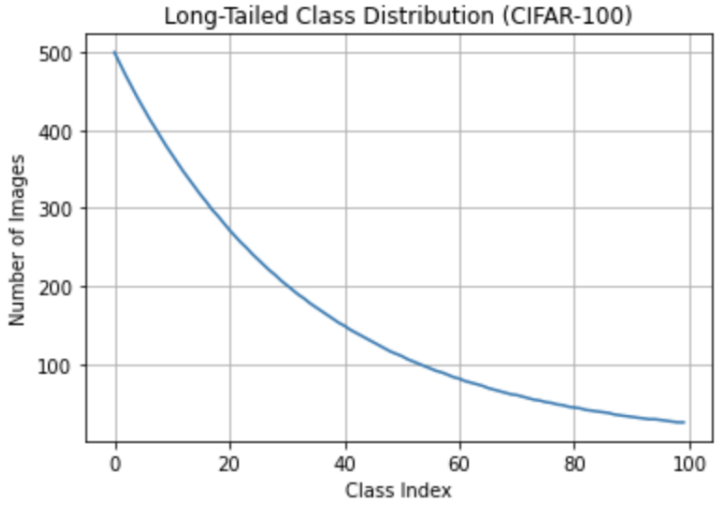

Figure 1: Class distribution plot of Long-Tailed CIFAR-100 dataset

2 Related Work

2.1 Overview of Existing Approaches

Long-tailed image classification has been an area of extensive research, with various old and new methods being applied to solve this problem. This section explores the contributions of recent works and their influence on our novel approach.

A **Systematic Review on Long-Tailed Learning** [6] has been done to compile the different techniques used to solve this problem. Each technique can be classified into one of eight categories, which are data balancing, neural architecture, feature enrichment, logits adjustment, loss function, bells and whistles, network optimization, and post hoc processing approaches. For our approach we will be mostly focusing on data balancing, neural architecture, as well as feature enrichment.

2.2 Data Balancing Approach

For data balancing, we are acting on the data we have directly, in the form of resampling, augmenting, or synthesizing. While resampling works to improve the impact the tail images have, the overall representation that is learned "may be of poor quality owing to the limited number of minority

samples." [7] The research done here demonstrated that data augmentation can help mitigate this issue by enriching the diversity and quantity of samples in the tail classes, ultimately leading to more robust and balanced feature representations. These encouraging results served as motivation for incorporating data augmentation into our framework as well.

Recent advances in generative modeling have introduced effective approaches to address data scarcity issues in imbalanced datasets. Our image generation approach builds upon two fundamental works in the field of image synthesis and processing.

The U-Net architecture introduced by Ronneberger et al. [8] was originally designed for biomedical image segmentation but has proven remarkably versatile for generative tasks. Its distinctive encoder-decoder structure with skip connections enables the preservation of spatial information throughout the network, allowing fine details to propagate from encoder to decoder. This characteristic is particularly valuable when generating synthetic samples for minority classes, as it helps maintain the distinctive visual features that define each class despite limited training examples.

Building upon architectural innovations like U-Net, Latent Diffusion Models proposed by Rombach et al. [3] refined image synthesis by performing diffusion in a compressed latent space rather than pixel space. This approach significantly reduces computational requirements while maintaining generation quality, making diffusion models more accessible for data augmentation tasks. While our implementation focuses on a lightweight diffusion model operating directly in pixel space to accommodate the 32×32 resolution of CIFAR-100 images, we adopt the core denoising process to generate diverse, high-quality samples for underrepresented categories.

2.3 Contrastive Learning Approach

Earlier, we mainly discussed the application of data balancing in order to improve classification on tail images; however, neural architecture design and feature enrichment strategies play a vital role in addressing the long-tailed distribution challenge as well. To explore this, Wang et al. propose a novel framework in their paper where they leverage the strengths of contrastive learning to enhance feature representations in imbalanced datasets [].

Rather than relying solely on a conventional convolutional neural network for classification, their system is composed of a hybrid loss framework: "a supervised contrastive loss to learn image representations and a cross-entropy loss to learn classifiers." This approach helps mitigate the bias toward head classes by forcing the model to learn more meaningful similarities between tail-class samples and their counterparts. The contrastive learning component, in particular, helps the model learn the feature space structure, making it more balanced and semantically aligned despite data scarcity in the tail.

Building on the idea of learning generalizable representations, the previous work primarily focused on using contrastive learning through a loss function to enhance feature learning. While effective, we wanted to take a different approach—one that would allow for more flexible and robust representation learning. Instead of relying solely on contrastive loss, we sought to explore a framework that learned latent space representations in a more encoder-driven format.

This led us to explore the DINO (Distillation with No Labels) framework for self-supervised learning [4]. DINO learns powerful image representations without requiring labeled data, which makes it particularly effective in scenarios with limited or imbalanced labels. By adopting an encoder-based learning approach, we push beyond the constraints of traditional contrastive learning and aim to produce richer, more robust latent feature representations. While the original DINO paper uses Vision Transformers (ViTs), we chose to implement DINO with a CNN to better suit our needs and computational considerations.

3 Methods

3.1 Overview

Our approach to addressing long-tail image classification on the CIFAR-100 dataset employs a comprehensive strategy combining architectural choices, data manipulation techniques, and advanced learning paradigms. We investigate **ResNet-50** as our baseline architecture, enhanced with various techniques to improve performance on minority classes. To tackle the inherent imbalance in the

dataset, we implement **traditional data augmentation** methods alongside **diffusion-based image generation** for minority classes. Additionally, we leverage **DINO (self-Distillation with NO labels)** as a contrastive learning framework to extract more robust features across classes regardless of sample frequency. Each method is evaluated independently and in combination to determine optimal strategies for long-tail classification.

3.2 Intuition

The challenge of long-tail distributions forces models to navigate a landscape where majority classes dominate the learning process at the expense of minority classes. Each component of our methodology addresses specific aspects of this challenge:

ResNet-50 provides a strong extraction backbone with sufficient capacity to learn characteristics across all classes. Its residual connections facilitate gradient flow, addressing the vanishing gradient problem that particularly affects learning from limited samples.

Traditional data augmentation (flipping, cropping, and color jittering) introduces variability within existing samples, helping prevent overfitting to the limited examples in minority classes. These transformations preserve semantic content while efficiently creating mass training instances.

Diffusion-based image generation targets the fundamental issue of data scarcity in tail classes. By learning the underlying distribution of each class independently, the diffusion model can synthesize diverse examples that maintain class-specific characteristics. This approach directly addresses the imbalance by filling in the long-tail with synthetic, but semantically valid samples.

DINO leverages contrastive learning principles to extract features that are invariant to class frequency. By learning from the data's structure through self-supervision rather than relying solely on labeled examples, DINO widens the representation gap between head and tail classes.

Together, these methods form a complementary approach that addresses long-tail classification at multiple levels: architectural design, data distribution manipulation, and feature learning paradigms.

3.3 ResNet

For the challenge of classifying images in the CIFAR-100 dataset we chose to go with a residual network. These networks are based on convolutional neural network architectures, but with the addition of a residual connection that mitigate the vanishing gradient problem on deeper networks.

For our implementation, we imported the pre-made ResNet-50 model from He et al. and applied it to our data [5]. The original model, however, is designed for images of size 224x224 instead of the 32x32 from CIFAR-100. As a result, the first convolution and pooling layer were modified to enhance the spatial features from our dataset. These modifications include changing the first layer to have a kernel size of three with a stride and padding of one along with removing the pooling layer.

With these modifications we then trained on the regular and long-tailed data to get base accuracies for both cases. To train, we passed in the training and validation set so that we could see results after each epoch. In both cases, we trained over 20 epochs, and used the cross-entropy loss function. Furthermore, our optimizer was Adam and was paired with a cosine annealing scheduler.

3.4 Data Augmentation

The first attempt to increase the classification accuracy in the tail classes was to perform data augmentation. Specifically, we used random cropping with padding to simulate slight variations in object positioning, followed by random horizontal flipping to account for possible left-right orientations of the objects. These augmentations help prevent the model from overfitting to specific image features and promote robustness to variations in object appearance. Finally, we normalized using dataset-specific mean and standard deviation values and converted the images to tensors for training. This combination of augmentation and preprocessing provides the model with additional data, allowing it to learn more generalized features for improved performance. We can now apply the augmented dataset over the original long-tailed dataset for retraining models, enabling them to better generalize to the tail classes and improve overall classification accuracy across both head and tail classes.

3.5 Diffusion Model for Image Generation

To address the challenge of long-tail classification, we implemented a generative approach using a diffusion model to synthesize additional images for under-represented classes. The diffusion model framework provides a controllable method for generating images that preserve the semantic characteristics of each class.

3.5.1 Diffusion Process Overview

The diffusion model operates on the principle of gradually adding Gaussian noise to images through a forward process and then learning to reverse this process. Formally, given an original image x_0, the forward diffusion process gradually adds noise over T timesteps according to a predefined variance schedule $\beta_1, ..., \beta_T$:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I})$$

This process can be expressed in a closed form for any timestep t:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I})$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha} = \prod_{i=1}^{t} \alpha_i$. At the final timestep T, the image x_t approaches a standard Gaussian distribution.

The reverse process then learns to gradually denoise from random Gaussian noise back to a clean image through a neural network that predicts the noise component:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

3.5.2 Network Architecture

Our implementation employs a lightweight U-Net architecture as the backbone for the diffusion model. The U-Net consists of:

1. **Encoder Path:** A series of residual blocks with two downsampling layers that progressively reduce spatial dimensions ($32 \times 32 \rightarrow 16 \times 16 \rightarrow 8 \times 8$) while increasing feature depth ($64 \rightarrow 128 \rightarrow 256$ channels). Each downsampling operation uses strided convolutions rather than pooling to maintain more spatial information.
2. **Bottleneck:** Two residual blocks that process the most compact representation at 8×8 spatial resolution with 256 channels, allowing the model to capture high-level semantic features of each class.
3. **Decoder Path:** Two upsampling layers with residual blocks that gradually restore spatial dimensions ($8 \times 8 \rightarrow 16 \times 16 \rightarrow 32 \times 32$) through transposed convolutions with stride 2. Each upsampling stage halves the number of channels ($256 \rightarrow 128 \rightarrow 64$).

The limited depth of our U-Net architecture (two downsampling/upsampling operations instead of the typical our or give used for higher-resolution images) is crucial for preserving the find details present in CIFAR-100's small 32×32 images. More aggressive downsampling would compress the spatial information too severely, leading to more blurry or less detailed generated samples.

Skip connections play a critical role in this architecture by directly connecting each encoder block to its corresponding decoder block. For the CIFAR-100 dataset, these connections are particularly important as they:

1. Maintain class-specific visual charateristics essential for generating recognizable objects
2. Facilitate gradient flow during training, which is especially important when training on minority classes with limited examples
3. Preserve spatial details that would otherwise be lost during the bottleneck compression

Each residual block incorporates group normalization followed by SiLU activation functions, providing stable training. The residual connections within each block allow the network to learn incremental modifications to the image features.

The time embedding module implements a sinusoidal position embedding that encodes the diffusion timestep into a high-dimensional vector. Based on the provided code, the embedding is calculated as:

1. First, a logarithmically spaced sequence of frequencies is created:

$$\text{freq} = \exp(-\log(10000) \cdot \frac{i}{d/2 - 1}$$

   for $i \in \{0, 1, ..., d/2 - 1\}$, where d is the embedding dimension.

2. Then, the embedding for a batch of timesteps t is computed by:

$$\text{emb}_{\sin}(t, i) = \sin(t \cdot \text{freq}_i)$$

$$\text{emb}_{\cos}(t, i) = \cos(t \cdot \text{freq}_i)$$

3. Finally, the sine and cosine components are concatenated:

$$\text{emb}(t) = [\text{emb}_{\sin}(t); \text{emb}_{\cos}(t)]$$

This time information is an adaptation of the positional encoding method introduced in the Transformer architecture [9]. For diffusion models working with CIFAR-100's small images, this embedding is crucial as it provides the network with precise information about the noise level at each denoising step, allowing for more controlled generation of fine details in the small images.

3.5.3 Training Procedure

The diffusion model was trained separately for each under-represented class in the CIFAR-100 dataset. For classes with fewer than 500 samples, ee employed the following training approach:

1. For each class, we collected all available training examples from the long-tail dataset.
2. The network was trained to predict the noise component added during the forward process, minimizing the mean squared error loss:

$$L = \mathbb{E}_{x_0, \epsilon, t} \left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2 \right]$$

   where ϵ is the actual noise added to create x_t, and ϵ_θ is the predicted noise from the neural network.

3. Training duration was adaptively determined based on the number of available examples in each class, with more epochs allocated to classes with fewer samples:

$$\text{num\_epochs} = \max(\text{min\_epochs}, \text{max\_epochs} \cdot (1 - n/n_{\text{target}}))$$

   where n is the number of examples in the class and n_{target} is the target number of examples (500 in our implementation).

3.5.4 Sampling Process

To generate new samples for minority classes, we employed the reverse diffusion process:

1. Start with random noise $x_T \sim \mathcal{N}(0, \mathbf{I})$
2. Iteratively denoise through timesteps $t = T, T-1, ..., 1$
3. At each step, compute:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right)$$

$$x_{t-1} = \mu_\theta(x_t, t) + \sigma_t \mathbf{z}$$

   where $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ and $\sigma_t^2 = \beta_t \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t}$. The final output x_0 represents a synthetic image for the target class.

For each class, we generated enough synthetic images to reach the target count of 500 examples per class, effectively transforming the long-tailed distribution into a more balanced one.

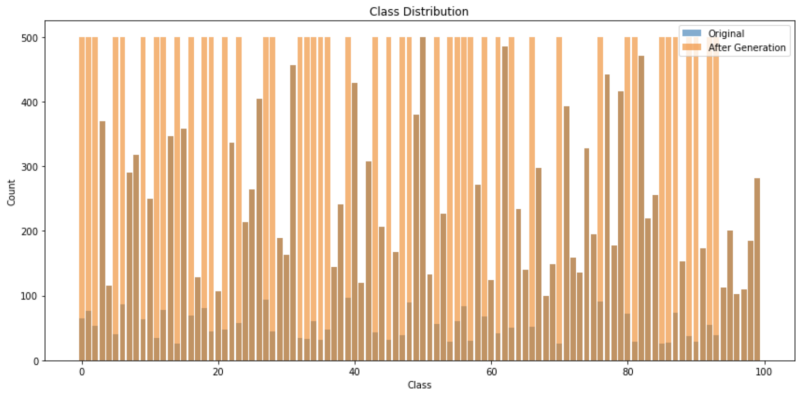

Figure 2: Class distribution plot of Generated CIFAR-100 dataset

3.5.5 Pipeline Integration

The generated images were combined with the original training set to create an augmented training dataset. This approach preserved all original training examples while supplementing minority classes with high-quality synthetic examples; see Figure [2].

By leveraging the diffusion model's ability to capture class-specific characteristics, the generated images maintain semantic consistency while introducing beneficial diversity, helping the classifier generalize better on minority classes without overfitting to the limited real examples.

3.6 DINO

Another approach to solve the issue of long-tail classification was through the use of DINO. This self-supervised learning method is designed to learn robust image representations without using labeled data, allowing for a better understanding of the semantics for each specific class.

3.6.1 DINO Overview

The model employs a teacher-student architecture, where the student network is trained to match the output of the teacher network. The teacher is not updated through gradient descent; instead, its weights are maintained as an exponential moving average (EMA) of the student's weights. This distillation mechanism helps the student network learn stable and more generalized features over time by mimicking the teacher's output.

A key component of DINO is the use of augmented views of the same image. For each input image, two different augmented versions are generated using a strong set of transformations, including random resized cropping, color jitter, random grayscale, and horizontal flipping. These augmentations are crucial—they ensure that the student and teacher are learning invariant features by seeing different "views" of the same image. These augmented images are then passed separately to the student and teacher networks, and their outputs are compared to encourage alignment.

The loss function used in DINO is a form of cross-entropy between softmax outputs, comparing the teacher's distribution p and the student's q, computed as:

$$\mathcal{L}_{\text{DINO}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C}\left(p_{ij}^{\text{teacher}} \cdot \log q_{ij}^{\text{student}}\right)$$

where the teacher output is computed using softmax with temperature T_t, and the student output is computed using log-softmax with temperature T_s:

$$p_{ij}^{\text{teacher}} = \frac{\exp(t_{ij}/T_t)}{\sum_{k=1}^{C}\exp(t_{ik}/T_t)}$$

$$q_{ij}^{\text{student}} = \log\left(\frac{\exp(s_{ij}/T_s)}{\sum_{k=1}^{C}\exp(s_{ik}/T_s)}\right)$$

Here, s_{ij} and t_{ij} represent the output logits of the student and teacher networks for input i and class j, C is the output dimension size, and N is the batch size. The use of temperature scaling sharpens or smooths the output distributions and plays a crucial role in stabilizing training. This formulation allows the student to learn invariant and generalized features by aligning its predictions with those of the teacher across differently augmented views of the same image.

3.6.2 Network Architecture

Our DINO framework is composed of two primary components: a backbone encoder and a projection head, each instantiated separately for the student and teacher networks. Both networks use a ResNet-18 architecture for the encoder, with key modifications. We replace the first convolutional layer with a smaller 3×3 kernel, as done earlier, to better capture spatial representations in small-resolution images. To adapt the encoder for self-supervised learning in DINO, we replace the final fully connected classification layer with an identity layer, effectively transforming the backbone into a pure feature extractor that outputs a 512-dimensional representation.

The extracted features are then passed to a projection head, defined as a Multi-Layer Perceptron (MLP). The head includes two linear layers with a GELU activation in between, followed by a final linear layer with weight normalization. The output dimensionality is set to 65,536 to match the softmax space used for representation alignment in the DINO loss.

The final linear layer uses weight normalization and has its norm parameter frozen during training to stabilize optimization. The teacher network is a frozen copy of the student network and is updated using an exponential moving average of the student weights

This architectural setup ensures that both networks evolve in sync, with the student gradually aligning its predictions to the more stable teacher over the course of training.

3.6.3 Training Procedure

The DINO framework was trained in two phases: self-supervised pretraining followed by supervised fine-tuning.

1. **Self-Supervised Pretraining**
   The student and teacher networks were trained using augmented image pairs with a contrastive loss, leveraging an exponential moving average to update the teacher network. The pretraining was performed for 20 epochs using the Adam optimizer.
2. **Supervised Fine-Tuning**
   After pretraining, the model was fine-tuned on the long-tailed dataset with a linear classifier. The model was optimized using Adam with a cosine annealing scheduler for 20 epochs.

This approach enabled the model to learn generalized features, which were then fine-tuned for the long-tailed classification task.

4 Experiments

4.1 Experimental Setup

4.1.1 Datasets

The experiments conducted in this work utilized variations of the CIFAR-100 dataset, which contains 60,000 color images (32×32 pixels) across 100 classes, with 500 training images and 100 testing images per class. We constructed the following dataset variations:

1. **Standard CIFAR-100**: The original evenly distributed dataset with 500 images per class, serving as out upper bound for performance comparison.

2. **Long-Tail CIFAR-100**: Our modified version with an exponential decay in sample count across classes, ranging from 500 images for head classes to as few as 25 images for tail classes. This represents our baseline for improvement.

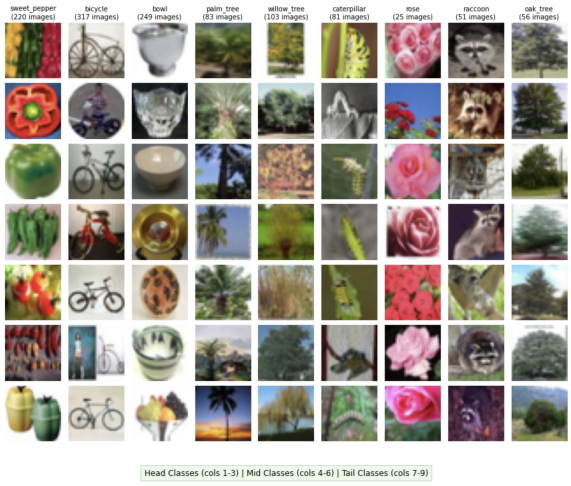

Figure 3: Long-Tail CIFAR-100 Distribution

3. **Augmented Long-Tail**: The long-tail dataset enhanced with traditional data augmentation techniques.

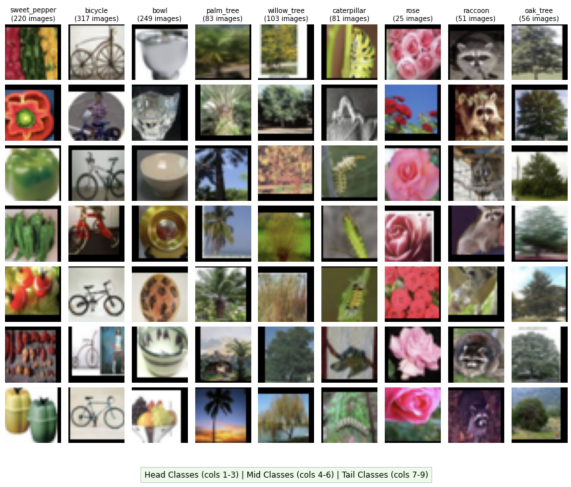

Figure 4: Augmented Long-Tail

4. **Diffusion-Generated Long-Tail**: The long-tail dataset balanced by adding synthetic images generated by our class-specific diffusion models, resulting in 500 images per augmented class (original + synthetic).

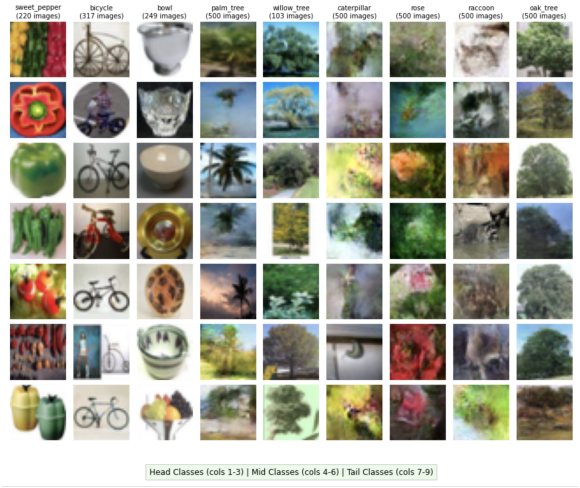

Figure 5: Diffusion-Generated Long-Tail

4.1.2 Evaluation Metrics

We evaluated model performance using overall accuracy as well as class-stratified accuracy to better understand the impact of our methods across the data distribution:

1. **Overall Accuracy**: The percentage of correctly classified images across the entire test set.
2. **Head-Class Accuracy**: Classification accuracy on the top 30 classes.
3. **Mid-Class Accuracy**: Classification accuracy on the middle 40 classes.
4. **Tail-Class Accuracy**: Classification accuracy on the bottom 33 classes.

This stratified analysis allows us to asses whether improvements are consistent across the distribution or primarily benefit specific portions of the long tail.

4.1.3 Model Configurations

We evaluated the following model configurations:

1. ResNet-50 Baseline: Standard ResNet-50 architecture trained on the long-tail dataset, serving as our primary baseline.
2. ResNet-50 + Standard Augmentation: ResNet-50 trained on the augmented long-tail dataset.
3. ResNet-50 + Diffusion Generation: ResNet-50 trained on the diffusion-enhanced long-tail dataset.
4. DINO: A smaller ResNet-18 architecture with DINO self-supervised learning framework, trained on the original long-tail dataset.
5. DINO + Standard Augmentation: DINO trained on the augmented long-tail dataset.
6. DINO + Diffusion Generation: DINO trained on the diffusion-enhanced long-tail dataset.

For reference, we also trained a standard ResNet-50 on the original evenly distributed CIFAR-100 dataset to establish the upper performance bound under ideal data conditions.

4.1.4 Training Details

All models were trained using a consistent protocol implemented in PyTorch and executed on NVIDIA A100-SXM4-40GB GPUs. The training process consisted of the following steps:

For each model configuration, we trained for 20 epochs using the Adam optimizer with an initial learning rate of 0.01 and a cosine annealing learning rate scheduler. We used a batch size of 32 for all configurations, with cross-entropy loss as our optimization objective. The training loop included regular validation checks to monitor performance across the head, middle, and tail classes.

4.2 Research Questions

Our research focuses on addressing the following key questions in the context of long-tail image classification:

- **Generation Quality:** Can diffusion models generate synthetic samples for tail classes that preserve class-specific visual characteristics while introducing beneficial diversity for classifier training?
- **Comparative Effectiveness:** How do different approaches (traditional augmentation, diffusion-based generation, and self-supervised learning) compare in addressing the inherent challenges of long-tail classification?
- **Feature Representation:** To what extent do contrastive learning frameworks like DINO improve feature representation understanding between head and tail classes, and how does this interact with data augmentation strategies?

These questions aim to evaluate the efficacy and limitations of the proposed approaches while providing insights for future research directions in imbalanced image classification scenarios.

4.3 Results

4.3.1 Quantitative Results

Our experiments evaluate the effectiveness of different approaches to addressing long-tail classification. Table 1 presents the test accuracy of all model configurations across the entire test set and each portion of the distribution (head, middle, and tail classes).

Table 1: Classification accuracy (%) on CIFAR-100 test set

Model Configuration	Overall	Head	Mid	Tail
ResNet-50 (Standard CIFAR-100)	53.10	48.77	55.89	53.77
ResNet-50 (Long-Tail)	29.50	46.79	31.28	9.25
ResNet-50 + Standard Augmentation	28.05	45.39	28.54	9.50
ResNet-50 + Diffusion Generation	**33.17**	**51.64**	**34.83**	11.87
DINO ResNet-18 (Long-Tail)	32.65	52.65	32.26	**13.85**
DINO ResNet-18 + Standard Augmentation	30.78	52.39	29.35	11.79
DINO ResNet-18 + Diffusion Generation	26.20	38.82	26.73	12.47

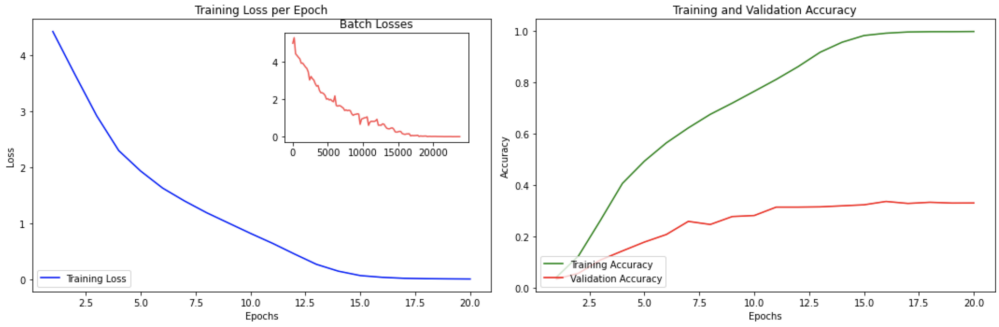

Figure 3: Figure 6: ResNet-50 Generated Images Loss/Val Plot

4.3.2 Analysis

The most notable finding from our results is the improvement achieved by the diffusion-based generation approach. When comparing the baseline ResNet-50 on the long-tail dataset (29.50% overall accuracy) with the ResNet-50 trained on the diffusion-enhanced dataset (33.17% overall accuracy), we observe a significant 3.67 percentage point improvement.

This improvement is particularly evident in the head and middle portions of the distribution, where accuracy increases of 4.85 and 3.55 percentage points were observed, respectively. For tail classes, which present the greatest challenge, diffusion generation provided a 2.62 percentage point improvement over the baseline (from 9.25% to 11.87%). This suggests that the synthetic images generated by our diffusion model successfully captured class-specific characteristics while introducing beneficial diversity to the training set.

Interestingly, while traditional data augmentation is a common approach to dataset enhancement, our experiments show that it actually resulted in a slight decrease in overall performance (from 29.50% to 28.05%). This indicates that simple augmentations like flipping and cropping may not provide sufficient diversity to address the fundamental issue of data scarcity in long-tail scenarios.

The DINO-based models demonstrate the potential of self-supervised learning approaches in addressing long-tail classification. The baseline DINO ResNet-18 achieved 32.65% overall accuracy, with particularly strong performance on tail classes (13.85%). This suggests that the contrastive learning framework helps extract more robust features that generalize better to underrepresented classes, even with limited examples.

Another important observation is the performance gap between models trained on the standard CIFAR-100 dataset and those trained on the long-tail variants. The standard ResNet-50 achieves 53.10% overall accuracy, substantially higher than any of our long-tail approaches. This highlights the inherent challenge of long-tail classification and suggests that while our methods provide significant improvements, there remains considerable room for further advancement toward the upper bound of performance.

In examining the relationship between head, middle, and tail class performance, we observe that diffusion-based generation helps normalize the accuracy distribution. While the baseline model shows a steep decline from head (46.79%) to tail (9.25%) classes, the diffusion-enhanced model exhibits a more gradual degradation, suggesting better feature learning across the entire distribution. This normalized performance profile indicates that synthetic data generation effectively addresses the training data imbalance.

5 Conclusion

5.1 Discussion

The experimental results demonstrate the efficacy of diffusion-based image generation for addressing long-tail classification challenges. By synthesizing additional samples for minority classes, we effectively "filled in" the tail of the distribution, leading to more balanced class representations and improved classification performance across the distribution spectrum.

While traditional data augmentation techniques failed to significantly improve performance, likely due to their limited capacity to introduce meaningful variation, our diffusion model succeeded in generating diverse yet class-consistent samples. This suggests that for extreme imbalance scenarios, robust generative approaches may be necessary to meaningfully expand the training distribution.

The DINO framework demonstrated promising results for tail classes, indicating that self-supervised learning can extract more generalizable features from limited data. However, the varying performance across different configurations underscores the complexity of long-tail classification and suggests that optimal approaches may depend on specific dataset characteristics and available computational resources.

In conclusion, our work demonstrates that diffusion-based image generation offers a viable solution for long-tail image classification, providing substantial improvements over baseline approaches.

5.2 Future Work

While our approach demonstrates promising results for long-tail image classification on the CIFAR-100 dataset, several opportunities for improvement remain. Our work reveals multiple directions for future research.

The current diffusion model implementation, while effective, requires separate training for each minority class. Future work could explore multi-class conditional diffusion models that can generate

samples across all classes with a single model. This would significantly reduce the computational overhead and training time while potentially allowing for better knowledge sharing across classes. Additionally, investigating efficient architectures that implement cross-attention mechanisms would make this approach more practical for larger datasets with more complex image resolutions.

Another promising direction is exploring hybrid approaches that combine the strengths of different data augmentation methods. For instance, integrating standard augmentation with image generation could yield more robust representations, particularly for extremely underrepresented classes.

The relationship between generation quality and classification performance also warrants deeper investigation. Understanding which aspects of generated images contribute most to improved classification could inform more targeted generation strategies. This includes exploring quality metrics beyond classification accuracy that better capture the semantic understandings of generated samples.

Finally, extending our approaches to larger, more complex datasets and real-world applications would validate their broader applicability. This includes adapting the methods to higher-resolution images, more diverse object categories, and naturally occurring long-tail distributions in domains such as medical imaging, remote sensing, and autonomous driving, where minority class recognition is particularly critical.

By addressing these areas, future research can build upon our findings to develop more effective, efficient, and broadly applicable solutions to the persistent challenge of long-tail image classification.

Acknowledgments

We thank the Northeastern University Deep Learning course team for guidance and support. Additional thanks to the authors of U-Net, DINO and related works for open-source resources.

References

[1] Krizhevsky, A. (2009). Learning Multiple Layers of Features from Tiny Images, , 32–33.

[2] Perez, L., Wang, J. (2017). The Effectiveness of Data Augmentation in Image Classification using Deep Learning. ArXiv:1712.04621 [Cs]. https://arxiv.org/abs/1712.04621

[3] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B. (2022). High-Resolution Image Synthesis with Latent Diffusion Models. ArXiv:2112.10752 [Cs]. https://arxiv.org/abs/2112.10752

[4] Caron, M., Touvron, H., Misra, I., Jegou, H., Mairal, J., Bojanowski, P., Joulin, A., Research, F. (n.d.). Emerging Properties in Self-Supervised Vision Transformers. https://arxiv.org/pdf/2104.14294

[5] He, K., Zhang, X., Ren, S., Sun, J. (2015). Deep Residual Learning for Image Recognition. https://arxiv.org/pdf/1512.03385

[6] A Systematic Review on Long-Tailed Learning. (2023). Arxiv.org. https://arxiv.org/html/2408.00483v1

[7] Ahn, S., Ko, J., Yun, S.-Y. (2023). CUDA: Curriculum of data augmentation for long-tailed recognition. ArXiv. https://arxiv.org/abs/2302.05499

[8] Ronneberger, O., Fischer, P., Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. https://arxiv.org/pdf/1505.04597

[9] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. (2017, June 12). Attention Is All You Need. ArXiv. https://arxiv.org/abs/1706.03762