Improved ResNet Model for CIFAR-10 Image Classification

Rahul Maliidi[1], **Nishant Sharma**[2], **Anushka Garg**[3]

NYU Tandon School of Engineering
rm7020@nyu.edu
ns6287@nyu.edu
ag10687@nyu.edu

Abstract

This report presents the implementation of an Improved ResNet model for image classification on the CIFAR-10 dataset. We demonstrated an effective balance between performance and model complexity by achieving a validation accuracy of 94.31% with 4.17 million parameters using customized ResNet topologies and optimal training techniques. In order to ensure efficient feature learning, our design used a progressive filter technique (32→64→128→192→256) with two residual blocks per layer and selective dropout (0.2-0.3). With the learning rate scheduler being crucial to convergence, training was optimized using SGD with Nesterov momentum (0.9), weight decay (0.0005), OneCycleLR scheduling, label smoothing (0.1), and gradient clipping. The model maintained high generalization and achieved 98.71% training accuracy. After training, predictions were generated for an unlabeled test dataset, and visualizations were implemented to analyze model behavior. Our findings highlight the effectiveness of deep residual networks combined with efficient training strategies for high-performance image classification.

Supporting Material

- **Code Repository:** Please refer to the attached link here for relevant codebase to the project.
- **Trained Model:** Please refer to the url attached here for our trained model submitted for the project.

Introduction

Computer vision has greatly improved because to deep learning, which allows machines to accurately identify and categorize images. Convolutional Neural Networks (CNNs), which extract hierarchical spatial characteristics from input images, are among the best architectures for image categorization. The vanishing gradient problem, in which gradients decrease as they move through layers and result in ineffective learning, is one of the difficulties CNNs encounter as they get deeper. Residual Networks (ResNet) address this problem by introducing skip connections, which let gradients avoid specific layers and allow deeper networks to train efficiently without experiencing a drop in performance.

The CIFAR-10 dataset, which comprises 60,000 color images in ten item categories, is used in this project to create an enhanced ResNet-18 model for image classification. Adaptive learning rate scheduling, regularization, and data augmentation are some of the sophisticated methods used to improve model performance. To enhance generalization, the dataset is transformed using techniques like normalization, random cropping, and horizontal flipping. Stochastic Gradient Descent (SGD) with Nesterov momentum is used to train the model, which reduces oscillations and speeds up convergence. Dropout layers, which randomly deactivate neurons during training, are also included at various stages to avoid overfitting.

The learning rate is dynamically changed during training using a OneCycleLR scheduler, which enables the model to experiment with a greater range of values before progressively arriving on an ideal learning rate. For steady training, gradient clipping is used to keep gradients from blowing up, and mixed precision training maximizes GPU memory utilization. Following training, an unlabeled test dataset is used to assess the model, and the predictions are saved in a CSV file for later examination. Additionally, a visualization feature is included to show test image samples and their anticipated labels. Building a strong and effective deep learning model that can correctly identify images while integrating strategies to improve generalization and computational efficiency is the aim of this research.

Existing Work

The CIFAR-10 dataset has continued to be a vital standard for assessing developments in image classification over the last four years, spurring breakthroughs in deep learning architectures and training methodologies. Deeper network training became possible with the advent of Residual Networks (ResNets), which solved the vanishing gradient issue. Building on this, DenseNet, which connects all layers in a feed-forward fashion, further decreased mistakes to 3.46%, while Wide Residual Networks (WRNs) enhanced performance by widening residual blocks, attaining an error rate of 4.0%. Neural Architecture Search (NAS) automated model design, resulting in error rates as low as 1.0%, and Shake-Shake regularization enhanced generalization, attaining an error rate of 2.86%.

A number of studies have improved image classification methods using CIFAR-10 between 2022 and 2025. In order to achieve 84.95% test accuracy, Yang et al. (2025)(3) developed an improved CNN architecture that integrated

batch normalization, dropout regularization, and deeper convolutional blocks. Rahman and Ozcan (2022)(5) presented a diffractive neural network-based time-lapse image classification technique that achieved 62.03% accuracy by utilizing lateral object movements for better categorization. Grønningsaeter et al.(4) demonstrated the promise of alternative learning models in 2024 by creating an improved image processing toolbox based on Tsetlin Machine composites, which achieved an accuracy of 82.8%. These developments significantly improve picture classification performance on CIFAR-10 by highlighting the ongoing growth of model designs, training methodologies, and automated learning approaches.

Dataset

In computer vision and machine learning research, the CIFAR-10 dataset—created by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton—is an often used benchmark. The 60,000 color photos, each with a 32 x 32 pixel resolution, are divided into 10 different classes: truck, airplane, car, bird, cat, deer, dog, frog, horse, and ship. There are 6,000 photos in each class, guaranteeing a fair representation in all categories.

A test set of 10,000 photos and a training set of 50,000 images make up the dataset. Five batches of 10,000 pictures each are created from the training data. Notably, the training batches may have minor differences in the distribution of classes, whereas the test batch is intended to have precisely 1,000 randomly chosen photos from each class. Together, the training batches guarantee that precisely 5,000 photos are used to represent each class.

To support supervised learning tasks, the images in CIFAR-10 have been painstakingly tagged and are taken from the larger "80 Million Tiny Images" dataset. The low-resolution photos in the dataset pose a special challenge, which makes it perfect for creating and testing algorithms that can identify intricate patterns.

Methodology

Data Augmentation

To enhance model generalization, we applied data augmentation to the CIFAR-10 training dataset. Our pipeline included random cropping (32×32 crops from 40×40 padded images) and random horizontal flipping to introduce geometric variance. Images were then converted to tensors and normalized using CIFAR-10's standard channel-wise mean $(0.4914, 0.4822, 0.4465)$ and standard deviation $(0.2023, 0.1994, 0.2010)$ values. These techniques effectively expanded our training set by creating varied versions of existing images, helping the model develop robustness to common image transformations.

Architecture

Our model implements an improved variant of the ResNet (Residual Network) architecture proposed by He et al. (1). The key innovation in ResNet is the use of residual connections to enable training of very deep networks by addressing the vanishing gradient problem. We have enhanced this architecture with strategically placed dropout layers and a progressive filter expansion strategy to achieve strong generalization while maintaining parameter efficiency.

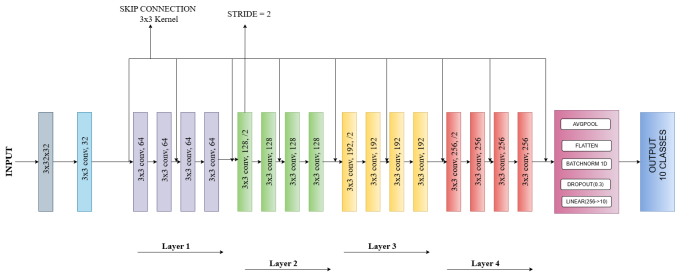

Figure 1: ImprovedResNet architecture with progressive filter expansion (32→64→128→192→256).

Basic Block Design Our implementation uses modified residual blocks where each block comprises:

- **First convolutional layer:** 3×3 kernel with stride 1 or 2 (for downsampling), followed by batch normalization, ReLU activation, and dropout (0.2)
- **Second convolutional layer:** 3×3 kernel with stride 1, followed by batch normalization
- **Shortcut connection:** Identity mapping when input and output dimensions match; 1×1 convolution with appropriate stride when dimensions change
- **Final activation:** ReLU applied after adding the shortcut connection to the residual path

The dropout layer (rate=0.2) after the first activation serves as a regularizer, reducing co-adaptation of feature detectors and improving generalization. The mathematical formulation of each residual block is:

$$y = F(x, \{W_i\}) + x \tag{1}$$

where $F(x, \{W_i\})$ represents the residual mapping learned by the two convolutional layers, and the addition operation is performed element-wise.

Network Architecture Our complete network consists of:

- **Initial stem:** A 3×3 convolutional layer with 32 filters (stride 1, padding 1), followed by batch normalization, ReLU activation, and dropout (0.25). Unlike the original ResNet, we omit max pooling to preserve spatial information for the small input images (32×32) in CIFAR-10.
- **Four stages with progressive filter expansion:**
  - **Stage 1:** Two basic blocks with 64 filters, maintaining spatial dimensions (32×32)
  - **Stage 2:** Two basic blocks with 128 filters, reducing spatial dimensions to 16×16 via strided convolution in the first block
  - **Stage 3:** Two basic blocks with 192 filters, reducing spatial dimensions to 8×8
  - **Stage 4:** Two basic blocks with 256 filters, reducing spatial dimensions to 4×4

This progressive filter expansion (32→64→128→192→256) provides a more gradual increase in capacity compared to the standard ResNet pattern (64→128→256→512), offering a better balance between representational power and parameter efficiency.

- **Classification head:** Global average pooling to convert the 4×4×256 feature maps to a 256-dimensional vector, followed by batch normalization, dropout (0.3), and a fully-connected layer that outputs logits for the 10 CIFAR-10 classes.

Weight Initialization and Regularization To enhance training stability and convergence, we implement specialized weight initialization strategies:

- **Convolutional layers:** Kaiming normal initialization (2) with fan-out mode and ReLU nonlinearity, which is particularly well-suited for networks with ReLU activations
- **Batch normalization layers:** Weights initialized to 1.0 and biases to a small positive value (0.01)
- **Fully connected layer:** Xavier uniform initialization to maintain variance across the network

For regularization, we employ a multi-tiered approach:

- **Dropout:** Three different rates at strategic locations (0.2 in residual blocks, 0.25 after initial convolution, 0.3 before classification)
- **Batch normalization:** Applied throughout the network to stabilize training and provide implicit regularization
- **Weight decay:** Applied during optimization to prevent overfitting

Model Complexity Our architecture contains 4,167,850 parameters distributed as follows:

- Initial convolutional layer: 864 parameters
- Stage 1 (64 filters): 94,720 parameters
- Stage 2 (128 filters): 378,240 parameters
- Stage 3 (192 filters): 910,848 parameters
- Stage 4 (256 filters): 2,273,280 parameters
- Classification head: 3,082 parameters

This parameter distribution concentrates capacity in the deeper layers, where more complex feature representations are learned. The total count of 4,167,850 parameters is well below the 5 million constraint

Training

Our training approach for the ImprovedResNet model prioritized both optimization stability and strong generalization. We employed SGD with momentum (0.9), Nesterov acceleration, and weight decay (0.0005). We trained the model for 120 epochs with a batch size of 64, balancing computational efficiency with optimization stability.

The OneCycleLR scheduler provided learning rate management, starting at 0.01, peaking at 0.1, and following a cosine annealing strategy. This scheduler completed a full cycle over the 120 training epochs, with 30% of the cycle

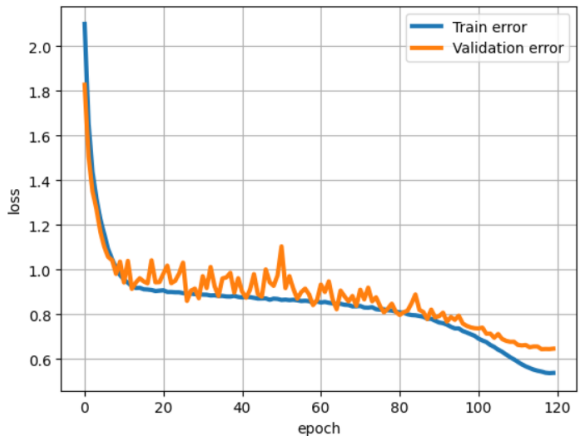

Figure 2: Training and validation loss

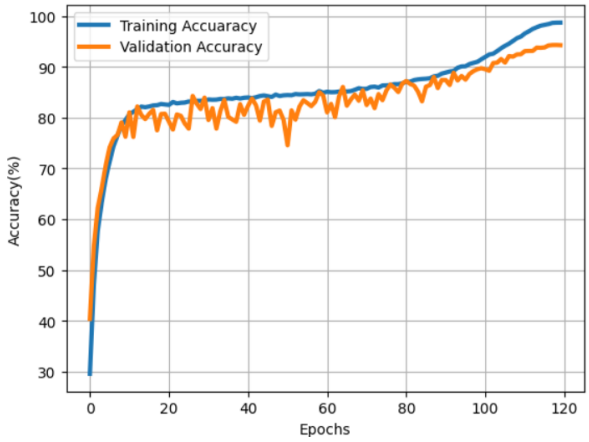

Figure 3: Training and validation accuracy

spent increasing the learning rate and the remainder dedicated to the cosine annealing phase. For regularization, we combined dropout layers at strategic positions, batch normalization, weight decay, and label smoothing (0.1) with our cross-entropy loss.

To accelerate training, we implemented mixed precision training with gradient clipping (max norm 1.0) to prevent instabilities. Early stopping with a patience of 10 epochs ensured optimal model selection. The model was trained on the CIFAR-10 dataset, consisting of 50,000 training images across 10 classes.

Our training process revealed three distinct learning phases as shown in the figures above. The final model achieved 98.71% accuracy on the training set and 94.31% on the validation set, demonstrating excellent generalization with minimal overfitting.

Results

Final Model Performance:

- Validation accuracy: 94.31%
- Test accuracy on Kaggle competition dataset: 83.059%

Training Process:

- Initial learning phase (Epochs 1-20): Rapid improvement from 29.55% to 82.59% training accuracy
- Steady improvement phase (Epochs 21-80): Gradual increase from 82.50% to 86.75% training accuracy
- Refinement phase (Epochs 81-120): Further enhancement reaching 98.71% training accuracy and 94.31% validation accuracy

Number of parameters: 4,167,850

Our model demonstrates excellent generalization on the CIFAR-10 validation set with a relatively modest parameter count. The strategic placement of dropout layers and the progressive filter expansion approach contributed to the model's ability to learn effectively while avoiding overfitting. The gap between training (98.71%) and validation accuracy (94.31%) indicates a well-regularized model that maintains good generalization capabilities.

The lower accuracy (83.059%) on the Kaggle competition dataset suggests a domain shift between the training distribution and the competition data, though the model still performs strongly on this unseen dataset.

Conclusion

In order to improve accuracy and generalization, this study used dropout regularization, progressive filter scaling, and sophisticated optimization approaches to offer an improved ResNet architecture for image classification on the CIFAR-10 dataset. The model successfully stabilized training and reduced overfitting by combining deeper convolutional layers, batch normalization, and strategic dropout integration, resulting in a validation accuracy of 94.31% while preserving computational economy. Stable convergence was made possible using the OneCycleLR scheduler with gradient clipping, which enabled the model to achieve 98.71% training accuracy while maintaining good generalization performance.

These improvements highlight how crucial it is to balance regularization, feature extraction power, and network depth in deep learning models. According to the investigation, robust classification accuracy requires a carefully planned architecture with tuned hyperparameters; merely increasing model complexity does not necessarily result in appreciable performance gains.

Future research will concentrate on using this improved ResNet model to more intricate datasets, including CIFAR-100, which has 100 categories and poses a more difficult classification problem. In order to adapt the architecture to large-scale datasets like ImageNet and enable its use in domain-specific tasks, transfer learning approaches will also be investigated. The model's practical relevance across a wider range of real-world picture classification problems will be further solidified by these expansions, which will assess scalability, generalization, and computational efficiency.

References

[1] He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep residual learning for image recognition*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 770-778).

[2] He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification*. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1026-1034).

[3] X. Yang, S. Yu, and W. Xu, "Enhanced convolutional neural networks for improved image classification," *arXiv preprint arXiv:2502.00663*, 2025. [Online]. Available: https://arxiv.org/abs/2502.00663

[4] Y. Grønningsæter, H. S. Smørvik, and O.-C. Granmo, "An optimized toolbox for advanced image processing with Tsetlin machine composites," *arXiv preprint arXiv:2406.00704*, 2024.

[5] M. S. Rahman and A. Ozcan, "Time-lapse image classification using diffractive neural networks," *arXiv e-prints*, 2022, arXiv:2208. [Online]. Available: https://arxiv.org/abs/2208.00000