# Optimizing Deep Learning Models for Image Classification on CIFAR-10

**ECE-GY 7123 Deep Learning**
**Amarnadh Reddy Mettu , DevaKumar Katta , Jeel Patel**

New York University Tandon School of Engineering
6 MetroTech Center
Brooklyn, New York 11201 USA

**Abstract**

This report focuses on optimizing a deep learning model for image classification on the CIFAR-10 dataset. We propose a modified ResNet architecture, leveraging techniques such as data augmentation and advanced optimization strategies to enhance model performance. The Adam optimizer with a learning rate scheduler is used to manage convergence efficiently. Our results show significant improvements in accuracy over the baseline ResNet model, demonstrating the effectiveness of our optimizations for image classification tasks. This work contributes to the development of efficient deep learning methodologies, offering a practical framework for similar applications.

## Code Availability

The source code for this project is available at: https://github.com/amarnadh145/deeplearning_mini/

## 1 Introduction

Image classification is a core task in computer vision, with the CIFAR-10 dataset serving as a widely used benchmark. Deep learning models, particularly ResNet architectures, have demonstrated strong performance on this dataset. However, increasing model complexity often leads to challenges such as overfitting, high computational costs, and inference latency.

This project aims to optimize a ResNet-based model for CIFAR-10, balancing accuracy and efficiency. We implement modifications to the standard ResNet architecture, leveraging PyTorch and GPU acceleration for faster training. To enhance generalization, we apply data augmentation techniques such as random cropping, horizontal flipping, color jittering, and random rotation. Additionally, we use the Adam optimizer with a learning rate scheduler to ensure stable convergence.

Our results demonstrate significant accuracy improvements over the baseline ResNet model while maintaining computational efficiency. This work provides insights into designing optimized deep learning models for image classification, contributing to more effective and practical implementations in real-world applications.



## 2 Methodology

This section explain the step by step process of the network, any techniques used and the hyperparameters chosen for the model.

### 2.1 Dataset

This project employs the CIFAR-10 dataset, a widely used benchmark for image classification. The dataset consists of 60,000 images of size pixels, categorized into 10 classes. The dataset is split as follows:

- Training Set: 50,000 images
- Test Set: 10,000 images (used for evaluation)
- Competition Test Set: `cifar_test_nolabel.pkl` containing unlabeled images

### 2.2 Data Preprocessing

To ensure consistency between training and inference, the following preprocessing transformations were applied using `torchvision.transforms`:

- Random Cropping (32, padding=4): Adds slight random cropping with padding to improve robustness.
- Random Horizontal Flip: Randomly flips images horizontally for data augmentation.
- Color Jitter: Randomly modifies brightness, contrast, saturation, and hue.
- Random Rotation (20 degrees): Adds random rotation up to 20 degrees to increase robustness.
- Random Affine Transformations: Randomly translates the image within a 10%range.
- Normalization: Standardizes pixel values using CIFAR-10 statistics:

$$\begin{aligned} \mu &= (0.4914, 0.4822, 0.4465) \\ \sigma &= (0.2023, 0.1994, 0.2010) \end{aligned} \tag{1}$$

## 3 Model Architecture

The ModifiedResNet model is based on the ResNet-18 architecture with key modifications aimed at optimizing performance on the CIFAR-10 dataset. Two variations of the architecture were developed to analyze the trade-off between model complexity and computational efficiency.

**Model 1: Deeper Modified ResNet** The first model is a deeper version of ResNet with an increased number of residual blocks, improving feature extraction and learning capacity. The architecture consists of:

- **Initial Convolution:** A 3x3 convolutional layer with 64 filters, stride=1, and padding=1.
- **Residual Blocks:**
  - **Layer 1:** Comprises 4 BasicBlocks operating on 64-channel feature maps, preserving spatial resolution while enhancing feature extraction.
  - **Layer 2:** Includes 4 BasicBlocks, increasing to 128 channels with the first convolution applying a stride of 2 to halve spatial dimensions.
  - **Layer 3:** Contains 3 BasicBlocks, expanding to 256 channels for high-level semantic feature extraction with additional downsampling.
  - **Trainable Parameters:** 4,697,162
  - **Memory Usage:** 36.56 MB

```
================================================================
Total params: 4,697,162
Trainable params: 4,697,162
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.01
Forward/backward pass size (MB): 18.63
Params size (MB): 17.92
Estimated Total Size (MB): 36.56
----------------------------------------------------------------
```

### 3.1 Model 2: Lighter Modified ResNet

This model follows the same general structure but with a reduced number of residual blocks, making it more lightweight and optimized for lower-resource environments.

- **Residual Blocks:**
  - **Layer 1:** 2 BasicBlocks operating on 64-channel feature maps, preserving spatial resolution while learning fundamental patterns.
  - **Layer 2:** 2 BasicBlocks processing 128-channel feature maps, introducing a stride of 2 in the first convolution to downsample the spatial dimensions.
  - **Layer 3:** 2 BasicBlocks increasing the depth to 256 channels, further refining high-level semantic representations required for classification.
- **Trainable Parameters:** 2,777,674
- **Memory Usage:** 21.11 MB

```
================================================================
Total params: 2,777,674
Trainable params: 2,777,674
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.01
Forward/backward pass size (MB): 10.50
Params size (MB): 10.60
Estimated Total Size (MB): 21.11
----------------------------------------------------------------
```

Figure 1: Visualization of parameters and memory usage for the lighter ResNet model.

## 4 Training

With the model architecture and hyperparameters set, the next step is to train the Modified ResNet on the CIFAR-10 dataset. The data is loaded through PyTorch's DataLoader, incorporating the above mentioned augmentation techniques to enhance generalization and prevent overfitting.

The Cross-Entropy Loss function is chosen due to its effectiveness in classification tasks. Given the categorical nature of the CIFAR-10 dataset, Cross-Entropy Loss ensures a well-calibrated prediction distribution over the 10 classes.

For optimization, the Adam optimizer is used due to its adaptive learning rate properties. To further refine training stability, ReduceLROnPlateau is employed as a learning rate scheduler, dynamically adjusting the learning rate when validation loss stagnates.

The model is trained for 75 epochs, and L2 regularization (weight decay) is applied to penalize large weights, further enhancing the model's generalization performance. This technique ensures that the network does not overfit to specific training examples but instead learns a robust representation that performs well on unseen test data.

## 5 Results and Discussion

Our optimized ResNet models were trained and evaluated on the CIFAR-10 dataset. The deeper Modified ResNet ([4,4,3]) achieved a final test accuracy of 93.20%, while the lighter Modified ResNet ([2,2,2,2]) obtained an accuracy of 93.11%.

### 5.1 Training Performance

The models were trained for **75 epochs**, with loss and accuracy monitored on both the training and test datasets. The performance of both models was analyzed to understand the impact of depth on classification accuracy and computational efficiency.

**Performance of Lighter Modified ResNet** The first model, a shallower variation of ResNet, was trained and evaluated on CIFAR-10. The training and testing loss curves for this model are depicted in Figure 2, while the corresponding accuracy trends are shown in Figure 3.


Figure 2: Train and test loss for the lighter ResNet model.

This model, with 2,777,674 parameters and a memory footprint of 21.11 MB, exhibited faster training times due to its reduced computational complexity.

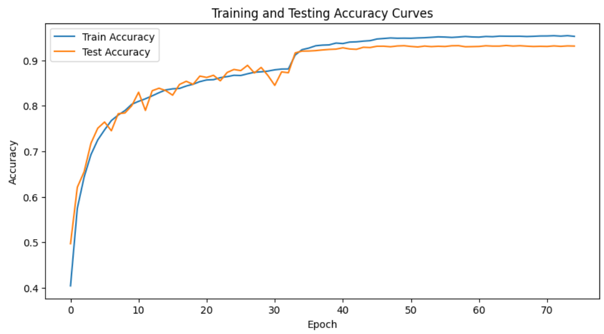


Figure 3: Train and test accuracy for the lighter ResNet model.

**Performance of Deeper Modified ResNet** The second model, a deeper variation of ResNet, was trained and evaluated using the same methodology to compare its performance against the lighter model. The training and test loss curves for this model are shown in Figure 4, and the corresponding accuracy trends are illustrated in Figure 5.

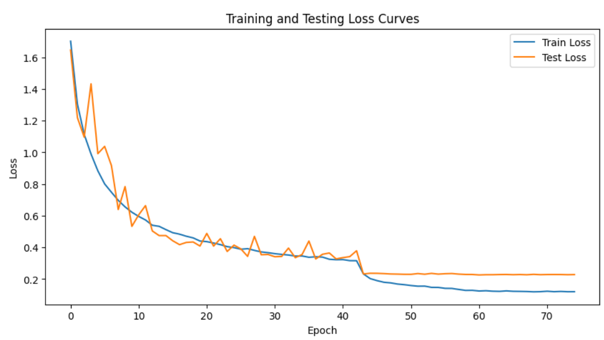


Figure 4: Train and test loss for the deeper ResNet model over 75 epochs.

The deeper model, which contains 4,697,162 trainable parameters and requires an estimated 36.56 MB of memory, achieved improved accuracy compared to the shallower model.

Overall, the study highlights the trade-off between model complexity and performance. The deeper model exhibits better classification accuracy.

## 6 Conclusion

In conclusion, our study demonstrates that with thoughtfully designed architectures and training strategies, it is possible to enhance the performance of deep learning models on benchmark datasets like CIFAR-10 without resorting to computationally intensive solutions. Our future endeavors will explore the application of the insights gained from this project to other computer vision tasks and datasets, as well as the incorporation of newer and more advanced neural network architectures.

The results highlight the trade-off between model depth and computational efficiency. The deeper model exhibited superior feature extraction, leading to slightly better accuracy, but required significantly more memory and had a higher parameter count.

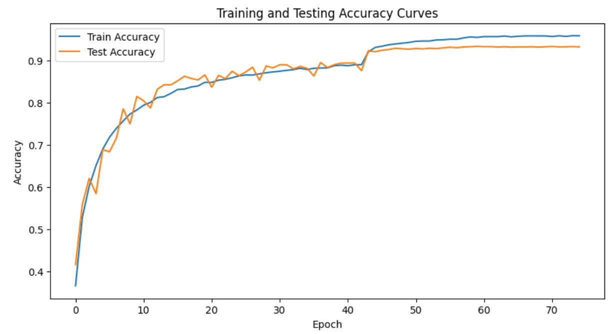


Figure 5: Train and test accuracy for the deeper ResNet model over 75 epochs.

## 7 Acknowledgments

We would like to thank Professor Chinmay Hegde for their guidance throughout this project. We also extend our gratitude to our peers at New York University for their valuable feedback and discussions. Lastly, we acknowledge the use of computational resources provided for training deep learning models efficiently.

## References

[1] K. He, X. Zhang, S. Ren, and J. Sun, ”Deep Residual Learning for Image Recognition,” in extitProceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[2] D. Shuvam, ”Implementation of ResNet Architecture for CIFAR-10 and CIFAR-100 Datasets,” 2023. Available: https://medium.com/@shuvamd210/unveiling-themetrics-mystery-exploring-scg-and-ccg-for-image-modelevaluation-76301445f583.

[3] K. Liu, ”PyTorch CIFAR Models,” 2021. Available: https://github.com/kuangliu/pytorch-cifar.