ResNet-Mini: A Custom Deep Architecture for CIFAR-10 Image Classification

Ali Aslanbayli, Farid Taghiyev

New York University
Tandon School of Engineering

Abstract

This paper presents a deep learning approach for image classification on the CIFAR-10 dataset. We designed a custom ResNet-inspired architecture, which we call ResNet-Mini, featuring progressive depth decrease and channel width expansion across network stages. Our model incorporates residual connections with mixed activation functions (GELU in the main path and ReLU after skip connections) to enhance gradient flow and learning dynamics. Through extensive experimentation with architectural choices and training strategies, including data augmentation techniques and learning rate scheduling, our model achieved 93.35% test accuracy while maintaining a relatively modest parameter count of approximately 5 million parameters. The progressive depth decrease strategy (starting with deeper early stages and gradually reducing depth in later stages) coupled with increasing channel width proved effective for balancing computational efficiency and classification performance. Our results demonstrate that carefully designed architectural choices can yield competitive performance without requiring excessive model complexity.

Introduction

Image classification is a fundamental task in computer vision with applications ranging from autonomous driving to medical diagnosis. The CIFAR-10 (Krizhevsky, Alex and Hinton, Geoffrey 2009) dataset, consisting of 60,000 32×32 color images across 10 classes, serves as a benchmark for evaluating image classification algorithms. This report presents our approach to developing a deep learning model for CIFAR-10 classification, detailing the architecture design, training methodology, and results achieved.

Our goal was to develop a high-performing yet computationally efficient model, balancing accuracy with model complexity. We implemented a custom ResNet-like (He et al. 2016) architecture with several optimizations to achieve state-of-the-art performance on this dataset.

GitHub Repo

The source code alongside all of the output files can be found by visiting https://github.com/aslanbayli/ResNetMini

Methodology

Model Architecture

We designed a custom deep residual network architecture, which we call ResNet-Mini. The architecture draws inspiration from the ResNet family but incorporates several modifications to improve performance on the CIFAR-10 dataset.

The key components of our architecture include:

- **Initial stem convolution**: Converting 3-channel images to 32 feature maps
- **Progressive depth decrease**: Starting with deeper layers (16 blocks) in early stages and decreasing depth ($8 \to 4 \to 2$ blocks) in later stages
- **Channel width expansion**: Gradually increasing channel width ($32 \to 48 \to 64 \to 152 \to 256$) to capture richer features
- **Residual connections**: Employing skip connections to mitigate the vanishing gradient problem
- **Mixed activation functions**: Using GELU (Hendrycks and Gimpel 2023) activations in the main path and ReLU after skip connections

The network consists of four main stages following the initial stem. Each stage processes feature maps at a particular spatial resolution, with downsampling occurring between stages. The final features are processed through adaptive average pooling and a fully connected layer for classification. Here is the architecture diagram of our neural network Fig. 1.

Residual Block Design

Our residual block implementation incorporates several design choices to enhance learning Fig. 2

The skip connection adapts to different input and output dimensions using 1×1 convolutions when needed. This design allows for efficient gradient flow throughout the network while maintaining representational capacity.

Training Strategy

We employed a comprehensive training strategy with the following components:

- **Optimizer**: AdamW (Loshchilov and Hutter 2019) with a learning rate of 0.0008

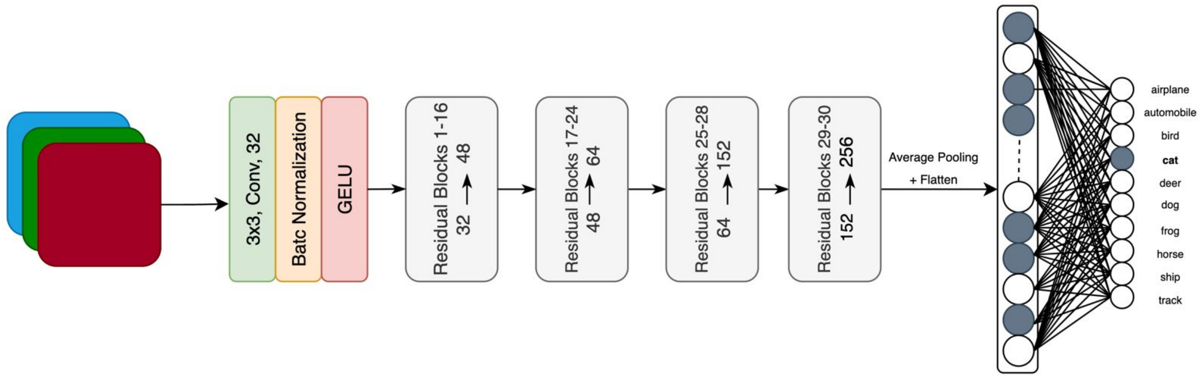

Figure 1: Arhcitecture of ResNet-Mini

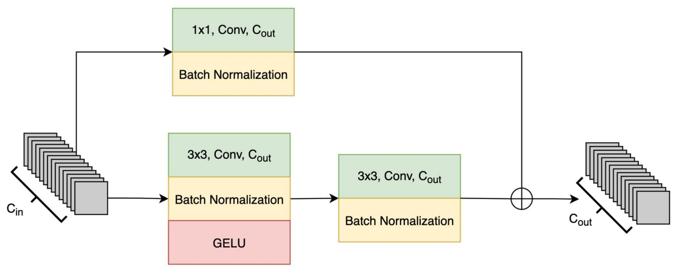

Figure 2: Residual Block

- **Learning rate scheduler**: CosineAnnealingLR with T_max=200
- **Batch size**: 64
- **Training epochs**: 60
- **Loss function**: Cross-Entropy Loss

Data Augmentation

To improve generalization, we implemented a robust data augmentation pipeline:

Listing 1: Transformations

```
Training Transforms:
- Random crop (32×32) with reflection
    padding of 4 pixels
- Random horizontal flip
- Random rotation (±10 degrees)
- Random affine transformations (shear=10,
    scale=0.8-1.2)
- Normalization (μ=[0.4914, 0.4822, 0.4465],
     σ=[0.2470, 0.2435, 0.2616])
```

For testing, we only applied normalization to maintain the integrity of the evaluation.

Results

Model Performance

Our ResNet-Mini model achieved impressive performance on the CIFAR-10 dataset:

- **Test Accuracy**: 93.35%
- **Training Accuracy**: 95.27%
- **Parameters**: 4,972,770 (approximately 5 million)

The performance metrics demonstrate the effectiveness of our architectural decisions and training methodology. Our model achieves competitive accuracy while maintaining a relatively small parameter count compared to larger models like ResNet-50 or DenseNet-121.

Training Dynamics

The training process showed consistent improvement in both training and testing accuracy. The model exhibited healthy learning behavior with a gradual reduction in the gap between training and testing performance, indicating good generalization.

Early in training (epochs 1-10), we observed rapid improvement in accuracy from 36.65% to 79.17%. The middle phase (epochs 11-40) showed continued but more moderate gains, while the final phase (epochs 41-60) demonstrated fine-tuning of performance, ultimately reaching 93.35% test accuracy. See Fig. 3.

Model Analysis

Analysis of our model reveals several interesting insights:

Depth vs. Width Trade-off: The progressive depth decrease coupled with width increase proved effective, allowing the model to capture fine details in early layers while extracting higher-level features in later stages.

Activation Functions: The combination of GELU (Hendrycks and Gimpel 2023) in the main path and ReLU after skip connections helped improve gradient flow and learning dynamics.

Computational Efficiency: Despite having approximately 5 million parameters, the model achieves high accuracy, demonstrating an effective balance between model capacity and computational requirements.

Discussion

Architecture Considerations

Our architectural decisions were guided by several considerations:

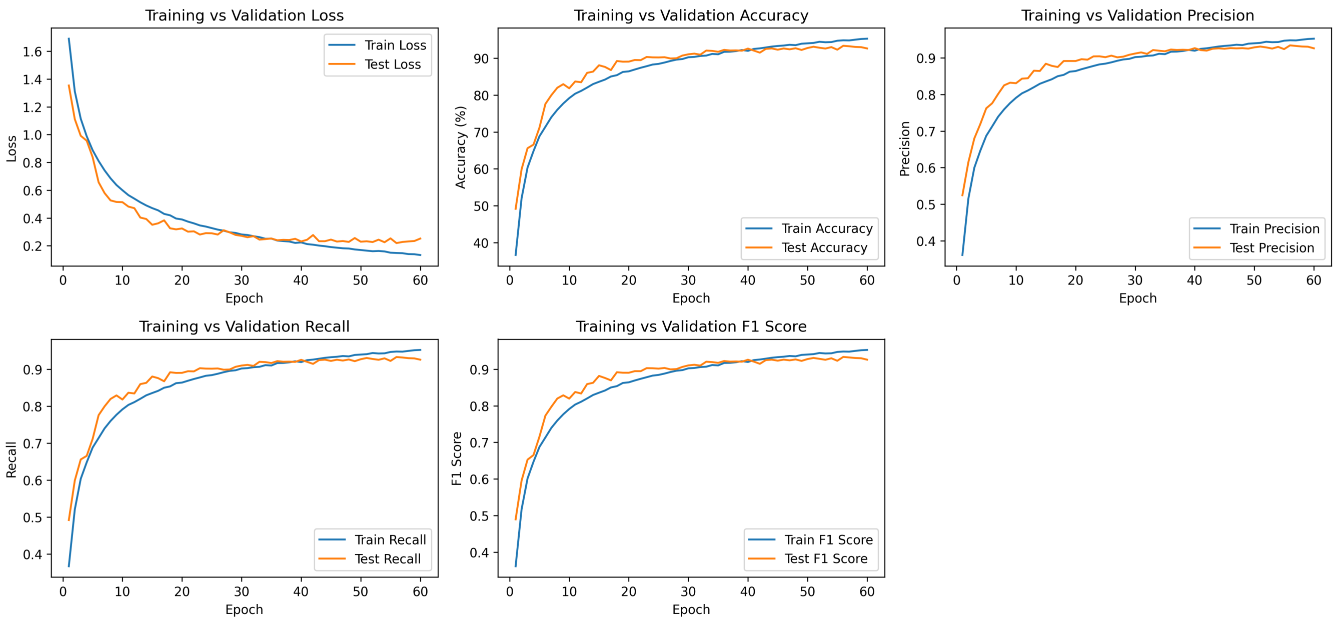

Figure 3: Training and validation metrics over epochs showing accuracy, loss, and F1-score.

- **Residual Learning**: The use of residual connections was crucial for training deeper networks, allowing gradients to flow more effectively during backpropagation.
- **Progressive Depth Decrease**: This design choice was motivated by the intuition that early layers benefit from more processing to extract low-level features, while later layers can be more parameter-efficient.
- **Channel Width Expansion**: Gradually increasing channel width allowed the model to develop increasingly abstract representations while maintaining computational efficiency.

Limitations and Future Work

Despite achieving strong performance, several limitations and opportunities for improvement exist:

- **Model Size**: While efficient for its performance, further optimizations could reduce parameter count through techniques like pruning or knowledge distillation.
- **Advanced Regularization**: Implementing techniques like Mixup or CutMix could potentially improve generalization further.
- **Architecture Search**: Automated architecture search methods could optimize the block distribution and channel widths more systematically.

Future work could explore these directions to push performance even further while maintaining or reducing computational requirements.

Conclusion

We presented a custom ResNet-Mini architecture for CIFAR-10 image classification that achieves 93.35% test accuracy with approximately 5 million parameters. Our approach demonstrates the effectiveness of residual learning, progressive depth decrease, and channel width expansion for this task.

The design principles explored in this work—balancing depth and width, employing mixed activation functions, and implementing effective data augmentation—provide valuable insights for developing efficient deep learning models for image classification tasks.

References

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Hendrycks, D.; and Gimpel, K. 2023. Gaussian Error Linear Units (GELUs). arXiv:1606.08415.

Krizhevsky, Alex and Hinton, Geoffrey. 2009. Learning multiple layers of features from tiny images. Technical report, Dept. of Computer Science, University of Toronto.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. arXiv:1711.05101.

External resources

This report was created with the help of Perplexity to make the expression of ideas more clear and structured.

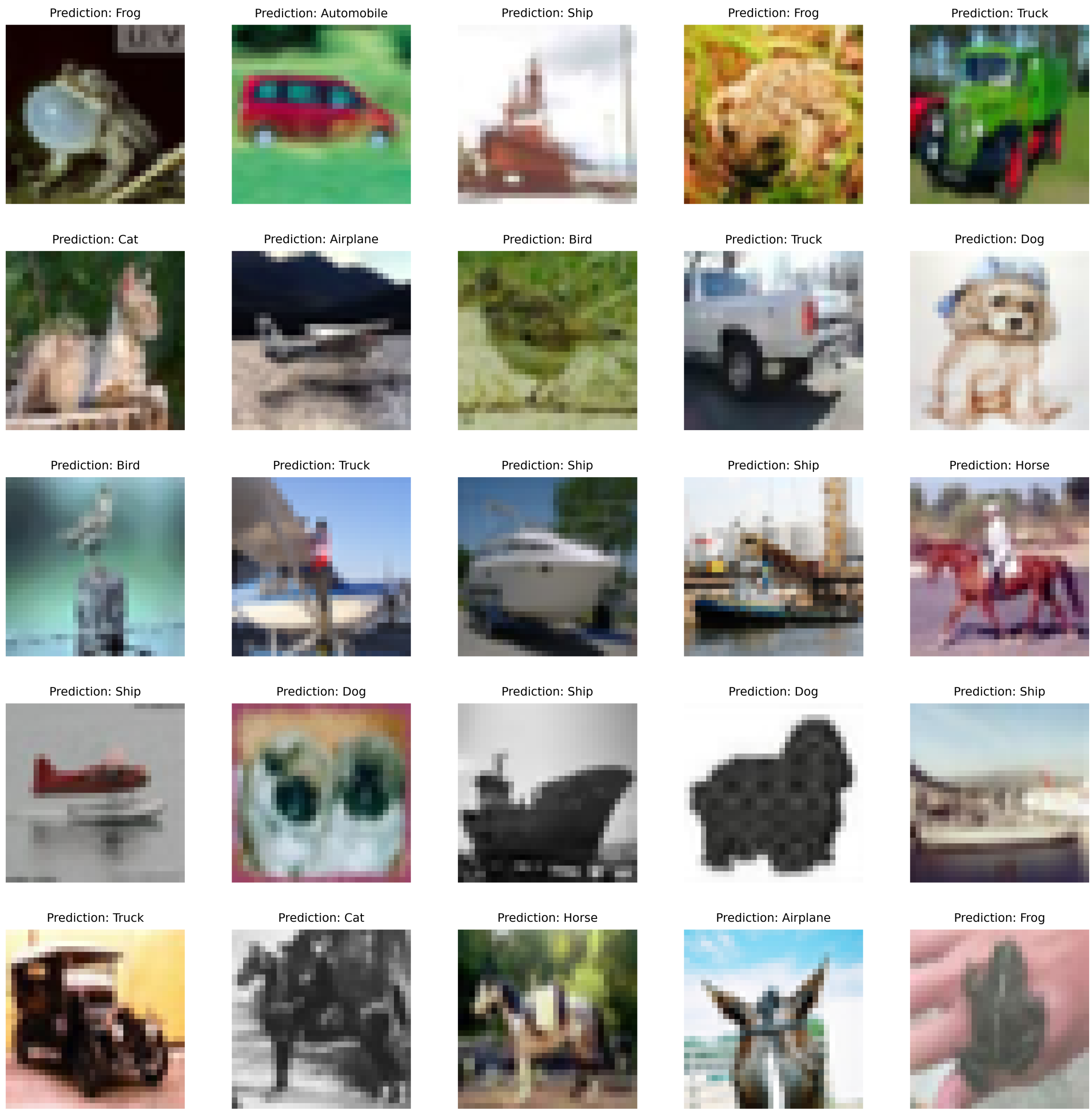

Figure 4: Model predictions on a random sample of CIFAR-10 test images.