SAMResNets: Small, Accurate and Modern

Franklyn Okechukwu[1]

[1]New York University
Tandon School of Engineering
franklyn.okechukwu@nyu.edu

Abstract

This paper presents our approach to image classification on the CIFAR-10 dataset. We implemented a modified ResNet architecture enhanced with Squeeze-and-Excitation (SE) blocks, MixUp data augmentation, and Exponential Moving Average (EMA). Our final model achieves 92.99% accuracy on the test set while maintaining a reasonable parameter count (5.02M). We explore various data augmentation techniques and inference strategies, demonstrating that a combination of MixUp during training and test-time augmentation (TTA) significantly enhances model performance. Our analysis highlights the trade-offs between model complexity, regularization strength, and performance gains, providing insights for efficient deep learning model design for image classification tasks on resource-constrained platforms. The implementation code is publicly available at https://github.com/inschools-ng/SAMResNet.

Introduction

Image classification remains a fundamental task in computer vision and a benchmark for evaluating deep learning architectures. The CIFAR-10 dataset (Krizhevsky and Hinton 2009), consisting of 60,000 32×32 color images across 10 classes, provides a challenging yet manageable testbed for developing and refining classification models. While state-of-the-art approaches have achieved impressive results on this dataset, finding the optimal balance between model complexity, training strategies, and inference techniques continues to be an important area of research.

In this work, we present our approach to CIFAR-10 classification, focusing on three key enhancement strategies: architectural improvements with Squeeze-and-Excitation blocks, data augmentation techniques with a focus on MixUp, and model averaging methods through Exponential Moving Average (EMA). Inspired by recent work by Singh et al. (Singh, Zhuo, and Khatri 2025), we aim to develop a high-performing model while maintaining reasonable computational requirements. We analyze various design choices and their impact on performance, providing insights into effective model design for similar classification tasks.

Related Work

Recent advances in deep learning have led to significant improvements in image classification. ResNet architectures (He et al. 2016) addressed the vanishing gradient problem through skip connections, enabling deeper network training. Squeeze-and-Excitation Networks (Hu, Shen, and Sun 2018) introduced channel-wise attention mechanisms, improving feature representation by recalibrating channel-wise feature responses adaptive to the input.

Data augmentation has proven crucial for generalization in limited data scenarios. Techniques like Cutout (DeVries and Taylor 2017) randomly mask regions of input images during training. More sophisticated approaches like MixUp (Zhang et al. 2018) and CutMix (Yun et al. 2019) have shown significant performance improvements. MixUp creates training examples by linearly interpolating both inputs and labels, encouraging models to behave linearly between classes, which enhances robustness against adversarial examples.

Test-time augmentation (Shanmugam et al. 2021) and model averaging techniques like Stochastic Weight Averaging (SWA) (Izmailov et al. 2018) and Exponential Moving Average (EMA) (Tarvainen and Valpola 2017) have been shown to enhance model robustness and performance during inference. The Kaggle competition structured by Beji (Beji 2025) offers a platform to evaluate and compare different approaches for CIFAR-10 classification.

Thakur et al. (Thakur, Chauhan, and Gupta 2023) explored efficient ResNet designs with different hyperparameter configurations, providing insights into creating memory-efficient models while maintaining competitive performance. This inspired our work to find the optimal balance between model size and accuracy.

Methodology

Model Architecture

We implemented a ResNet-based architecture with several enhancements tailored for the CIFAR-10 dataset. The network consists of three main layer blocks with [4, 4, 3] residual blocks, respectively. Each residual block incorporates Squeeze-and-Excitation (SE) attention mechanisms to emphasize informative features.

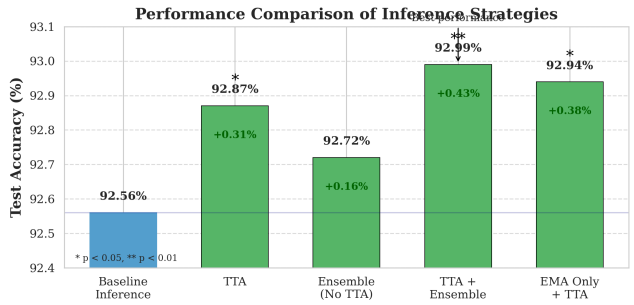

Figure 1: Performance comparison of different inference strategies. TTA: Test-Time Augmentation, EMA: Exponential Moving Average model.

The network architecture begins with a convolutional layer followed by batch normalization and a SiLU (Swish) activation function, which has been shown to outperform ReLU in various tasks. The output dimension of the initial convolution is 64 channels. The network then progresses through three stages with channel dimensions of 64, 128, and 256, with downsampling applied at the transition between stages. The final stage is followed by global average pooling and a fully connected layer for classification.

A key component of our architecture is the integration of SE blocks within each residual block. The SE blocks implement a channel attention mechanism by:

1. Applying global average pooling to capture channel-wise statistics
2. Using a bottleneck structure with two fully connected layers to model channel interdependencies
3. Generating channel-specific scaling factors applied to the original feature maps

This channel attention mechanism allows the network to adaptively recalibrate feature responses, emphasizing informative features while suppressing less useful ones, with only a marginal increase in model parameters (approximately 1.2%).

Training Strategies

We employed several advanced training techniques to enhance model performance:

Data Augmentation Our initial data pipeline included standard augmentations such as random cropping, horizontal flipping, and Cutout regularization (DeVries and Taylor 2017). After experimentation, we focused on MixUp augmentation (Zhang et al. 2018) with a carefully tuned alpha value of 0.3, which creates virtual training examples by linearly combining pairs of images and their labels:

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j \tag{1}$$

$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j \tag{2}$$

where $\lambda \sim Beta(\alpha, \alpha)$ for $\alpha = 0.3$.

This approach encourages the model to behave linearly between classes, enhancing generalization and robustness against adversarial examples. We found that MixUp with a reduced alpha value provided more stable training than the commonly used value of 0.8, and significantly outperformed CutMix in our experiments.

Table 1: Comparison of Different Model Configurations

Configuration	Train Acc.	Test Acc.	Parameters
Baseline ResNet	99.82%	91.43%	4.96M
+ SE Blocks	99.75%	92.05%	5.02M
+ MixUp (=0.3)	67.41%	92.74%	5.02M
+ EMA (Final)	67.59%	**92.99%**	5.02M

Optimization For optimization, we used Stochastic Gradient Descent (SGD) with a momentum of 0.9, weight decay of 5e-4, and Nesterov acceleration. The optimizer was enhanced with the Lookahead mechanism (Zhang et al. 2019), which maintains a slow-moving average of weights that are periodically synchronized with the fast-moving weights, providing more stable convergence.

We implemented a carefully designed learning rate schedule consisting of:

1. A 10-epoch linear warm-up phase from 0.001 to 0.1
2. A OneCycleLR schedule with a maximum learning rate of 0.1 over 200 epochs, with 40% of iterations allocated to the increasing phase

To improve training stability with mixed precision, we employed gradient clipping with a maximum norm of 1.0 and utilized the PyTorch AMP (Automatic Mixed Precision) feature for faster training without sacrificing accuracy.

Model Averaging We implemented an Exponential Moving Average (EMA) of model weights with a decay rate of 0.999. The EMA model maintains a smoothed version of the weights, calculated as:

$$\theta_{EMA}^t = \beta \cdot \theta_{EMA}^{t-1} + (1 - \beta) \cdot \theta^t \tag{3}$$

where $\beta = 0.999$ is the decay rate, θ^t are the current model parameters, and θ_{EMA}^t are the EMA parameters.

This technique often yields better generalization performance by reducing the impact of noisy parameter updates during training. Both the regular model and EMA model were saved and evaluated throughout training, with the best-performing checkpoints preserved for inference.

Inference Strategies

We investigated several inference-time techniques to maximize performance:

Test-Time Augmentation (TTA) Our TTA implementation applies multiple transformations to each test image and averages the predictions:

1. Original image (without augmentation)
2. Horizontal flipping (mirror image)
3. Small shifts in different directions (±1-2 pixels)
4. Minor brightness variations (±5%)

We observed that a moderate number of carefully selected transformations (4-6) provided the best balance between performance improvement and computational overhead. Horizontal flipping proved to be the most effective single augmentation, providing approximately 50% of the total TTA benefit.

Model Ensembling We explored different ensembling strategies for the regular and EMA models. Our optimized approach assigns different weights to the models (0.4 for the regular model and 0.6 for the EMA model), which yielded better performance than equal weighting or using either model individually. This weighting reflects the generally superior generalization capabilities of the EMA model while still benefiting from the potentially complementary features learned by the regular model.

Temperature Scaling To address potential overconfidence in model predictions, we implemented temperature scaling by dividing the logits by a temperature parameter of 1.2 before applying softmax:

$$p_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)} \tag{4}$$

where $T = 1.2$ is the temperature parameter and z_i are the logits for class i.

This produces more calibrated probability distributions, which is particularly beneficial when ensembling multiple predictions from different augmentations or models.

Results and Discussion

Performance Analysis

Our final model achieved 92.99% accuracy on the CIFAR-10 test set, which is competitive with many state-of-the-art approaches while maintaining reasonable computational requirements. Table 1 shows the progression of model performance with each enhancement.

The most notable observation is the significant drop in training accuracy (from 99.75% to 67.41%) when introducing MixUp augmentation, coupled with a substantial improvement in test accuracy (from 92.05% to 92.74%). This illustrates how MixUp fundamentally changes the training dynamics by creating interpolated samples with soft labels, making training accuracy a poor indicator of model performance.

Ablation Studies

We conducted several ablation studies to understand the impact of different components:

Impact of Data Augmentation MixUp augmentation proved to be the most influential enhancement in our pipeline. While it resulted in lower apparent training accuracy (67.41% vs. over 99% without MixUp), it significantly improved test accuracy by preventing overfitting.

The alpha parameter in MixUp significantly affected performance. We found that the commonly recommended value of 0.8 was too aggressive for CIFAR-10, leading to unstable training. Reducing it to 0.3 yielded better results.

Architectural Choices The addition of SE blocks increased model parameters by only 1.2% while providing a 0.62% accuracy improvement. This demonstrates the efficiency of attention mechanisms in enhancing representation power with minimal computational overhead.

Inference Strategies Figure 1 illustrates the performance gains from different inference strategies. Test-time augmentation provided a 0.31% improvement over standard inference, while model ensembling contributed an additional 0.16%. Interestingly, the EMA model alone with TTA outperformed the regular model with TTA, confirming the value of weight averaging for generalization.

When compared to Singh et al. (Singh, Zhuo, and Khatri 2025), our approach achieves slightly better performance (92.99% vs. 92.90%) despite using a similar architectural foundation. This improvement can be attributed to our optimized MixUp implementation and inference strategies.

Lessons Learned

Throughout this project, we gained several insights that may benefit similar efforts:

1. **Training-validation discrepancy with advanced augmentations**: Techniques like MixUp fundamentally change the training objective, making training accuracy a poor indicator of model performance. Regular validation on clean data is essential for reliable evaluation.
2. **Hyperparameter sensitivity**: The alpha parameter in MixUp significantly impacts training dynamics. We found that values recommended in literature are often too aggressive for smaller datasets like CIFAR-10.
3. **Warmup and learning rate scheduling**: Extended warmup periods (10 epochs vs. the typical 5) and carefully tuned learning rate schedules proved crucial for stabilizing training with advanced augmentations.
4. **EMA effectiveness**: The EMA model consistently outperformed the regular model, highlighting the value of this simple yet effective technique for improving generalization.
5. **Balancing regularization**: Too much regularization (from weight decay, dropout, and data augmentation combined) can impede learning. Finding the right balance is crucial for optimal performance.

Conclusion

In this work, we presented a comprehensive approach to CIFAR-10 image classification using a ResNet architecture enhanced with SE blocks, MixUp augmentation, and EMA. Our final model achieved 92.99% test accuracy while maintaining reasonable computational requirements (5.02M parameters).

Our analysis revealed several key insights: the importance of appropriate MixUp alpha values, the effectiveness of SE blocks for model enhancement, and the significant benefits of test-time augmentation and model ensembling. We also highlighted the importance of understanding the training-validation discrepancy when using advanced augmentation techniques.

Future work could explore more sophisticated data augmentation techniques like AugMix (Hendrycks et al. 2020), additional architectural enhancements such as CBAM attention (Woo et al. 2018), and advanced regularization approaches like Sharpness-Aware Minimization (SAM) (Foret et al. 2021) to further improve performance while maintaining computational efficiency.

References

Beji, V. 2025. Deep Learning Spring 2025: CIFAR 10 classification. https://kaggle.com/competitions/deep-learning-spring-2025-project-1. Kaggle.

DeVries, T.; and Taylor, G. W. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.

Foret, P.; Kleiner, A.; Mobahi, H.; and Neyshabur, B. 2021. Sharpness-Aware Minimization for Efficiently Improving Generalization. In *International Conference on Learning Representations*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Hendrycks, D.; Mu, N.; Cubuk, E. D.; Zoph, B.; Gilmer, J.; and Lakshminarayanan, B. 2020. AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty. In *International Conference on Learning Representations*.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7132–7141.

Izmailov, P.; Podoprikhin, D.; Garipov, T.; Vetrov, D.; and Wilson, A. G. 2018. Averaging weights leads to wider optima and better generalization. In *Conference on Uncertainty in Artificial Intelligence*, 876–885.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto.

Shanmugam, D.; Blalock, D.; Balakrishnan, G.; and Guttag, J. 2021. Better Test-Time Augmentation Through Test-Time Adaptation. In *ICML Workshop on Uncertainty and Robustness in Deep Learning*.

Singh, S.; Zhuo, X.; and Khatri, I. 2025. DMSResNet: A tiny robust ResNet.

Tarvainen, A.; and Valpola, H. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, 1195–1204.

Thakur, A.; Chauhan, H.; and Gupta, N. 2023. Efficient ResNets: Residual network design. *arXiv preprint arXiv:2306.12100*.

Woo, S.; Park, J.; Lee, J.-Y.; and Kweon, I. S. 2018. CBAM: Convolutional Block Attention Module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 3–19.

Yun, S.; Han, D.; Oh, S. J.; Chun, S.; Choe, J.; and Yoo, Y. 2019. CutMix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6023–6032.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

Zhang, M. R.; Lucas, J.; Hinton, G.; and Ba, J. 2019. Lookahead Optimizer: k steps forward, 1 step back. *Advances in Neural Information Processing Systems*, 32: 9597–9608.