Enhancing Image Classification with Modified ResNet Architectures

Md Abdur Rahman Fahad, Mahmood Ahmed
Department of Computer Science
Missouri State University
Springfield, Missouri, USA
{mf8494s,ma597s}@missouristate.edu

***Abstract*—Deep convolutional neural networks have achieved remarkable success in image classification, but training very deep models still presents challenges due to optimization difficulties such as vanishing gradients. Residual Networks (ResNets) addressed these challenges through skip connections that stabilize gradient flow. But the skip connection in standard ResNets is static and treats all identity features equally, regardless of their relevance. In this work, we propose two simple modifications to the ResNet architecture that introduce learnable and dynamic gating mechanisms in the skip connections. These gated connections allow the model to selectively control how much information from the identity path should be preserved. Experiments on CIFAR-10 and CIFAR-100 using ResNet-18 and ResNet-34 demonstrate that our gated variants improve classification performance with minimal computational overhead. The results highlight that simple modifications to skip connections can enhance the performance with similar efficiency of standard ResNet architectures.**

***Index Terms*—Image Classification, ResNet, Gating, Deep Learning**

I. Introduction

Image classification is a fundamental problem in computer vision with applications in autonomous driving, medical imaging, content filtering, etc. Deep convolutional neural networks (CNNs) have significantly advanced this task.

A pivotal moment in this advancement came with the introduction of AlexNet [1], which achieved a landmark victory in the 2012 ImageNet challenge by effectively leveraging GPUs and introducing key architectural features like ReLU activation and dropout. Following this initial success, early architectures such as VGG [2] demonstrated that increasing network depth could substantially improve recognition accuracy even further.

VGG relied on deep stacks of convolutional layers, but simply adding more layers often leads to optimization difficulties such as vanishing gradients or unstable updates. As depth increases, earlier layers struggle to learn discriminative features, leading to performance degradation despite higher model capacity.

An alternative approach focused on computational efficiency and multi-scale feature extraction. This came with the introduction of GoogLeNet (also known as Inception v1) [3], which was the winner of the ILSVRC 2014 challenge. The core innovation of GoogLeNet was the "Inception module," an architecture that operates with kernels of different sizes (1x1, 3x3, and 5x5 convolutions, along with max pooling) in parallel across the same input layer, concatenating the results to capture features at multiple scales. This design allowed the network to go both wider and deeper (22 layers) while maintaining a low computational budget compared to VGG by heavily utilizing 1x1 convolutions for dimensionality reduction and replacing fully connected layers with global average pooling. GoogLeNet also incorporated auxiliary classifiers during training to inject gradients deeper into the network, which helped mitigate the vanishing gradient problem.

Residual Networks (ResNets) introduced by He et al.[4] solved this challenge by using skip connections that allow gradients to bypass layers. The skip connection preserves the identity mapping, enabling deeper architectures with over 100 layers to be trained successfully. Skip connections changed the learning objective from mapping $H(x)$ to learning a residual function $F(x) = H(x) - x$.

However, the skip pathway in standard ResNets is fixed: the output of the residual block is always combined with the identity input using simple addition. This raises an important question: *Should all identity information always be passed forward equally*?

In this project, we propose adding gating mechanisms to the skip connection to make the identity pathway learnable and adaptive. Our hypothesis is that allowing the network to modulate the identity flow can improve performance without significantly increasing model complexity.

II. Dataset

Experiments were performed on the CIFAR-10 and CIFAR-100 benchmark datasets, both consisting of 32×32 RGB images with a fixed resolution and consistent preprocessing pipeline. CIFAR-10 contains 10 coarse-grained object categories, whereas CIFAR-100 includes 100 fine-grained classes grouped into 20 superclasses, making it significantly more challenging due to higher inter-class similarity and fewer samples per class. Each dataset is partitioned into 50,000 training images and 10,000 test images.

III. Methodology

A. Models

We adopt the standard ResNet-18 and ResNet-34 architectures as implemented in our project codebase. Each block consists of:

- Two 3×3 convolutions with BatchNorm and ReLU
- A skip connection that adds the input to the output

Formally, a standard residual block computes:

$$H(x) = \mathrm{ReLU}(F(x) + x)$$

In our implementation, ResNet-18 uses a **(2, 2, 2, 2)** block configuration, meaning each of the four stages contains 2 BasicBlocks. ResNet-34 is deeper with a **(3, 4, 6, 3)** layout, placing 3, 4, 6, and 3 BasicBlocks in the four stages, respectively.

B. Proposed Modification: Gated Skip Connections

We introduce two gating mechanisms to extend the residual block.

1) Approach 1: Learnable Scalar Gate: A learnable parameter λ, initialized to 1 for stability, scales the identity connection:

$$H(x) = F(x) + \lambda x$$

Here, λ is a per-channel learnable tensor implemented as:

$$\lambda \in \mathbb{R}^{1 \times C \times 1 \times 1}$$

This approach preserves the simplicity of the original block while introducing adaptability.

2) Approach 2: Dynamic Convolutional Gate: We use a 1×1 convolution followed by a sigmoid activation to compute an input-dependent gating mask:

$$H(x) = F(x) + \sigma(W_g * x) \odot x$$

This allows the model to dynamically decide which identity features to emphasize or suppress based on the input.

C. Training Pipeline

The training script uses the following components:

- Optimizers: Adam, SGD, AdamW
- Loss Function: Cross-Entropy Loss
- Epochs: 100
- Data Augmentation: random crop, horizontal flip, color jitter, and rotation

D. Hyperparameter Optimization

Automated hyperparameter tuning was performed using Optuna [5]. Each trial trained the model for 20 epochs, and the objective function returned the validation accuracy, which Optuna maximized. The hyperparameter search space was defined as follows:

- **Learning rate:** $[10^{-5}, 10^{-1}]$
- **Weight decay:** $[10^{-6}, 10^{-2}]$
- **Optimizer:** {SGD, Adam, AdamW}
- **Batch size:** {64, 128, 256}
- **Augmentation strategy:** {none, standard, heavy}
- **Momentum (SGD only):** $[0.7, 0.99]$

All trials used the same training pipeline, data preprocessing steps, and validation protocol to ensure a fair comparison. After running multiple optimization rounds, the final configuration was selected based on the highest validation accuracy.

Figures 1 and 2 show the hyperparameter importance computed by Optuna for ResNet-18 and ResNet-34, respectively. In both models, learning rate dominated the importance distribution. Which means that the model's performance is highly sensitive to the step size of gradient updates. Batch size and weight decay contributed moderately for ResNet-18, but for ResNet-34, their impact remained minimal. Optimizer choice and augmentation strategy had comparatively low influence.

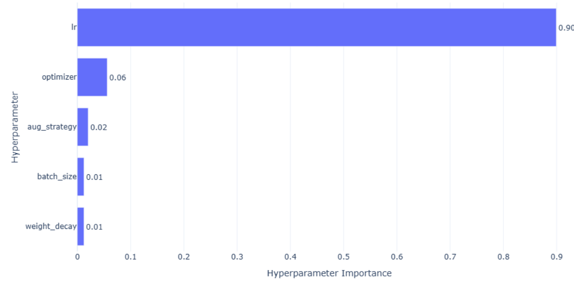

Fig. 1. Hyperparameter importance for ResNet-18

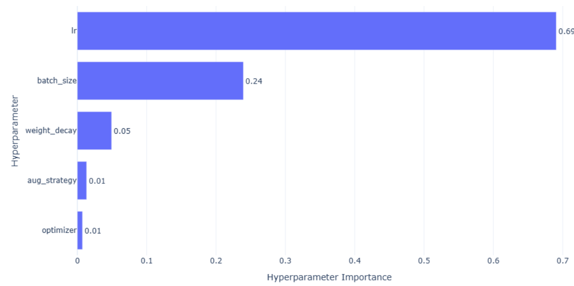

Fig. 2. Hyperparameter importance for ResNet-34

Figures 3 and 4 illustrate the objective value (validation accuracy) across training steps for all trials. Some of the runs were pruned by optuna as they did not show promising results.

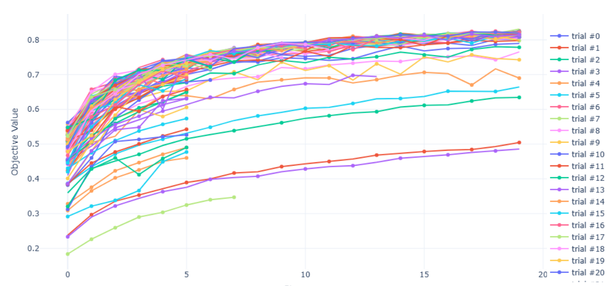

Fig. 3. ResNet-18 trial-wise learning curves showing variation in convergence behavior across different hyperparameter configurations.

The best hyperparameters obtained were:

- **ResNet-18:**
  optimizer = AdamW,
  learning rate = 0.001055,
  batch size = 64,
  augmentation = standard,
  weight decay = 4.65×10^{-5}.

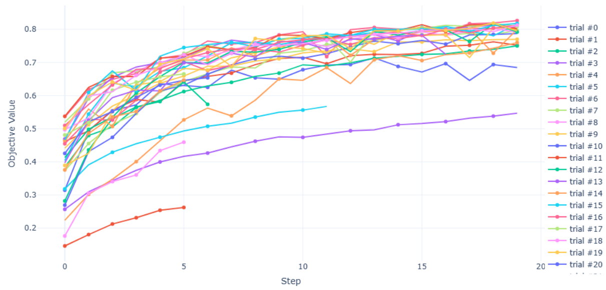

Fig. 4. ResNet-34 trial-wise learning curves, where most trials converge to a similar performance band, indicating robustness of the architecture.

- **ResNet-34:**
  optimizer = AdamW,
  learning rate = 0.001011,
  batch size = 64,
  augmentation = standard,
  weight decay = 2.57×10^{-4}.

IV. Experiments

Experiments on CIFAR-10 and CIFAR-100 show that both gated architectures outperform the standard ResNet-18 and ResNet-34 baselines.

A. Training

The training and validation loss curves for ResNet18 and ResNet34 (Figures 9 and 11) show stable convergence across all model variants. The Gated1 and Gated2 approaches maintain comparable or slightly lower training loss compared to the baseline, indicating that the gating mechanisms helped in the training process. Validation loss remains smooth for both architectures, with the gated variants demonstrating reduced oscillations in deeper layers, suggesting improved regularization behavior.

The corresponding accuracy curves (Figures 5 and 7) also follow consistent learning trajectories. All models reach strong performance within the first 30 epochs, with gated models showing marginally accelerated early learning. For ResNet34, the Gated1 variant achieves slightly higher final training and validation accuracy. The gated models generally sustain more stable validation accuracy late in training, indicating that the learnable skip-connections helped reduce overfitting in deeper architectures.

Overall, these training curves confirm that the introduction of gating into the residual connection can in fact improve convergence smoothness and generalization, specially in the deeper ResNet34 architecture.

We additionally report the training and validation losses for the ResNet-18 variants on CIFAR-100 in Figure 10, and for the ResNet-34 variants in Figure 12. Similarly, the corresponding training and validation accuracies are shown in Figure 6 for ResNet-18 and Figure 8 for ResNet-34.

B. Analysis of Learned Gating Behavior

To better understand how the proposed Gated1 mechanism influences the residual pathways, we analyzed the distribution

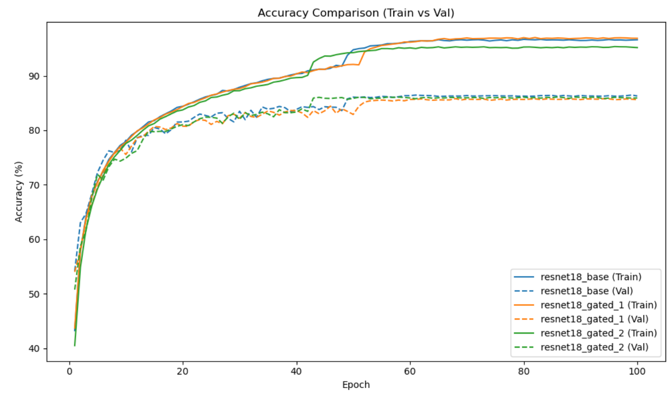

Fig. 5. Accuracy Curve – ResNet18(CIFAR-10)

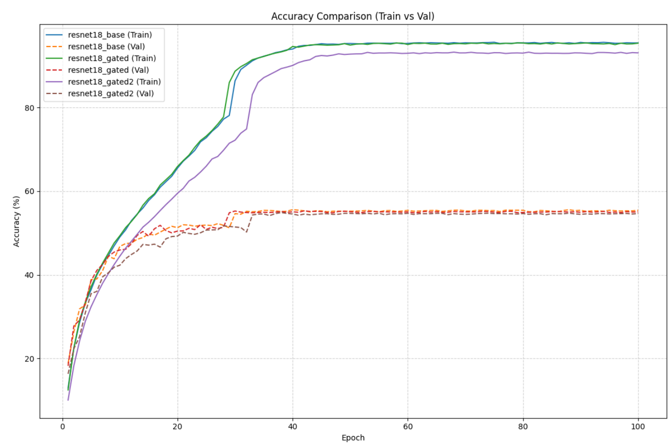

Fig. 6. Accuracy Curve – ResNet18(CIFAR-100)

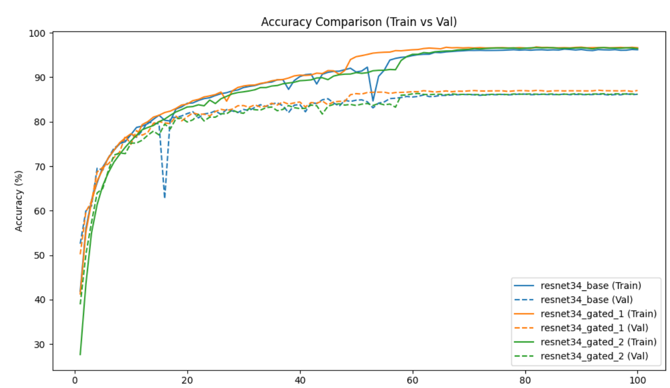

Fig. 7. Accuracy Curve – ResNet34(CIFAR-10)

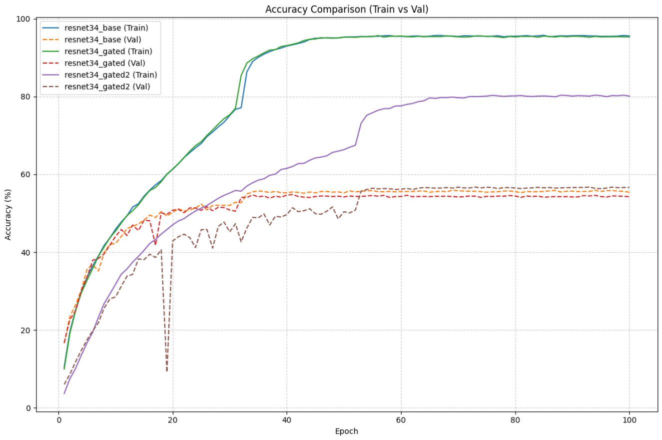

Fig. 8. Accuracy Curve – ResNet34(CIFAR-100)

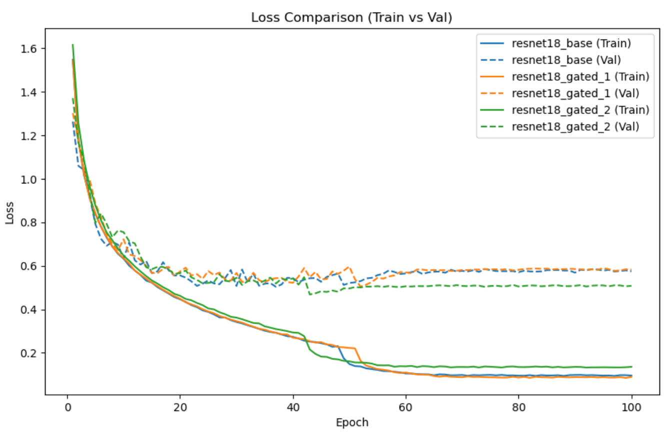

Fig. 9. Loss Curve – ResNet18(CIFAR-10)

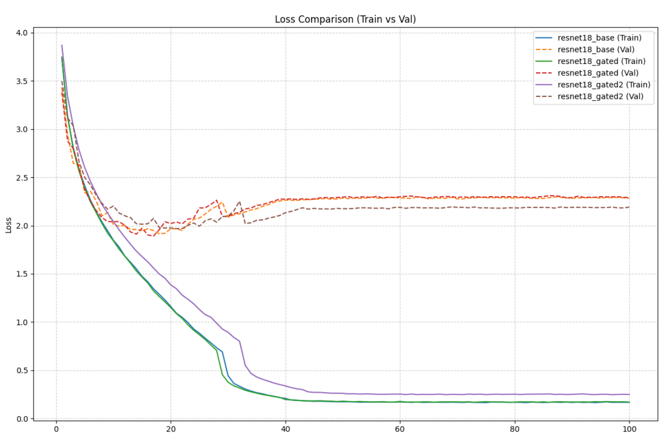

Fig. 10. Loss Curve – ResNet18(CIFAR-100)

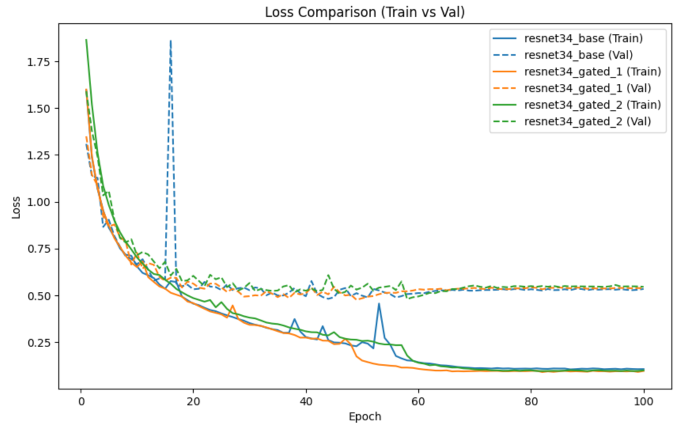

Fig. 11. Loss Curve – ResNet34(CIFAR-10)

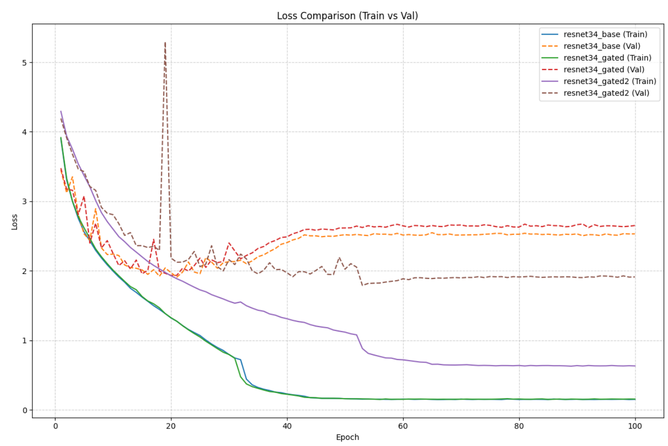

Fig. 12. Loss Curve – ResNet34(CIFAR-100)

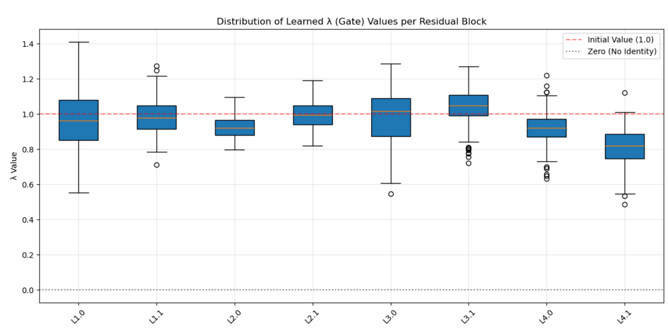

Fig. 13. Distribution of learned λ values for ResNet18

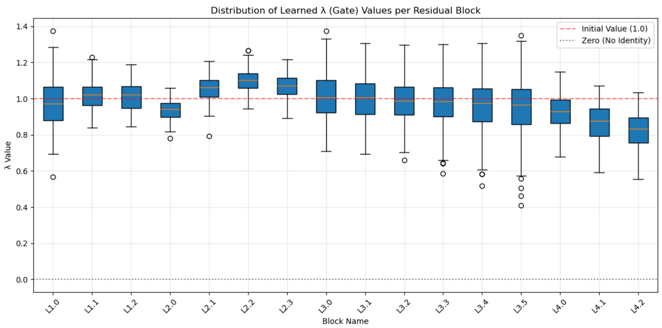

Fig. 14. Distribution of learned λ values for ResNet34

of the learned λ values across all residual blocks in ResNet18 and ResNet34. As shown in Figure 13, the ResNet18 model learns λ values that generally remain close to the initialization point of 1.0 but, several blocks show $\lambda < 1$, suggesting that the network learned to suppress the skip connection when necessary.

A similar trend is observed in the deeper ResNet34 model in Figure 14, where earlier layers maintain values near 1 while deeper blocks show a wider spread of λ values. This indicates more aggressive gating in later stages of the network. None of the gates were close to zero, confirming that the identity connection remains necessary.

Overall, these results show that the Gated1 architecture does not simply scale the identity uniformly, instead, it learns specific adjustments that enhance flexibility while preserving stability of residual learning.

Lastly, we report the final training and validation accuracies after training for both CIFAR10 and CIFAR100 in figure I, for CIFAR100 we reported top 1 accuracy.

TABLE I
FINAL ACCURACIES ON CIFAR-10 AND CIFAR-100

	CIFAR-10		CIFAR-100	
Model Name	**Train Acc**	**Val Acc**	**Train Acc**	**Val Acc**
Resnet18_base	0.9649	0.8720	0.9545	0.5545
Resnet18_gated1	0.9497	0.8696	0.9538	0.5506
Resnet18_gated2	0.9686	0.8714	0.9307	0.5467
Resnet34_base	0.9440	0.8690	0.9556	0.5550
Resnet34_gated1	0.9542	0.8716	0.9531	0.5425
Resnet34_gated2	0.9507	0.8670	0.8010	0.5663

C. Test Results

V. COMPUTATIONAL COMPLEXITY ANALYSIS

To understand the efficiency impact of the proposed gated skip connections, we compare the parameter count and computational cost in FLOPs of all model variants using Fvcore [6] developed by FAIR(Facebook Artificial Intelligence Research). As shown in Table IV, both gating approaches introduce only minimal overhead relative to the baseline ResNet architectures. The dynamic convolutional gate (Gated2) increases the parameter count by less than one million and adds

TABLE II
RESULTS ON CIFAR-10

Model Name	Accuracy	Precision	Recall	F1-Score
Resnet18_base	0.8647	0.8647	0.8647	0.8645
Resnet18_gated1	0.8581	0.8586	0.8581	0.8583
Resnet18_gated2	0.8612	0.8613	0.8612	0.8611
Resnet34_base	0.8626	0.8630	0.8626	0.8628
Resnet34_gated1	**0.8709**	**0.8716**	**0.8709**	**0.8710**
Resnet34_gated2	0.8635	0.8637	0.8635	0.8635

TABLE III
RESULTS ON CIFAR-100

Model Name	Top 1 Acc	Top 5 Acc	Precision	Recall	F1-Score
Resnet18_base	0.5560	0.8134	0.5567	0.5560	0.5549
Resnet18_gated1	0.5531	0.8143	0.5540	0.5531	0.5523
Resnet18_gated2	0.5491	0.8137	0.5520	0.5491	0.5494
Resnet34_base	0.5591	0.8187	0.5591	0.5591	0.5577
Resnet34_gated1	0.5480	0.8113	0.5466	0.5480	0.5454
Resnet34_gated2	**0.5673**	**0.8298**	**0.5679**	**0.5673**	**0.5663**

only a small FLOP increase, which is negligible compared to the overall cost of ResNet-34. These results demonstrate that the proposed gating mechanisms enhance model performance while preserving computational efficiency.

VI. QUALITATIVE ANALYSIS OF MODEL PREDICTIONS

We provide a qualitative alanysis of the model's predictions on CIFAR-10 and CIFAR-100. Figure 15 presents correctly classified CIFAR-10 samples, where the model assigns high confidence to the true class and maintains a clear margin from other categories. Also, Figure 16 shows misclassified CIFAR-10 examples, where many errors are from visually similar class images such as truck–automobile or bird–ship. In several cases the top-1 prediction is incorrect, but the true label still appears among the top-5 probabilities.

A similar result can be seen in CIFAR-100, shown in Figures 17 and 18. Correctly classified examples show high confidence and misclassifications often occur in fine-grained categories where small visual differences are difficult to capture in low-resolution images.

VII. CONCLUSION

We introduced two variants of gated skip connections for ResNet architectures. Both approaches enhance the model's ability to refine identity mappings, improving classification performance on CIFAR datasets with minimal increment in

TABLE IV
PARAMETER COUNT AND FLOPS COMPARISON

Model Architecture	Params (M)	FLOPs (G)
ResNet18 Base	11.18	1.82
ResNet18 Gated1	11.18	1.82
ResNet18 Gated2	11.88	1.92
ResNet34 Base	21.29	3.67
ResNet34 Gated1	21.29	3.67
ResNet34 Gated2	22.55	3.88

computational cost. Our findings prove that even simple modifications to skip connections can enhance deep network performance. Future work may explore expanding the modifications on deeper Resnet models and train on larger datasets like Imagenet.

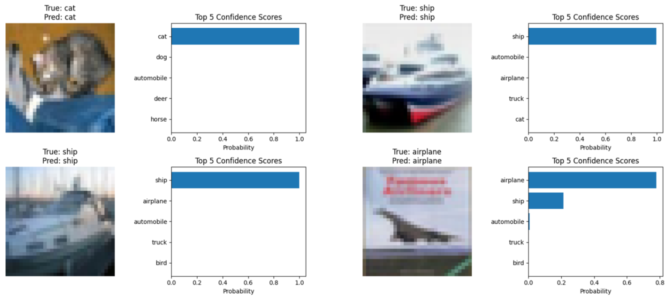

Fig. 15. Correctly classified examples on CIFAR-10

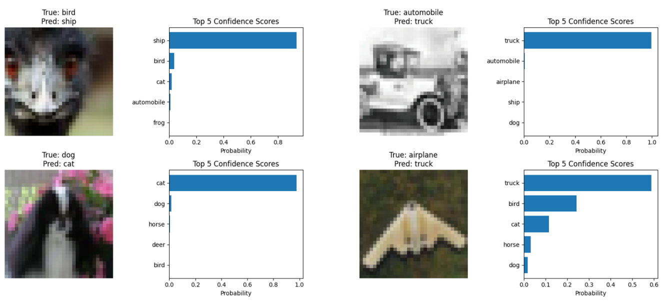

Fig. 16. Incorrectly classified examples on CIFAR-10

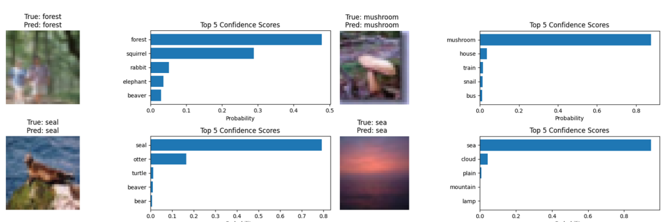

Fig. 17. Correctly classified examples on CIFAR-100

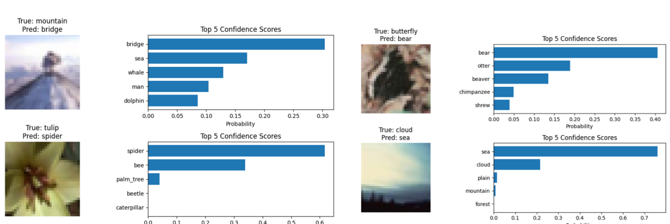

Fig. 18. Incorrectly classified examples on CIFAR-100

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In NeurIPS 2012

[2] Karen Simonyan, and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014).

[3] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going Deeper with Convolutions. In CVPR 2015

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In CVPR, 2016, pp. 770-778

[5] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19). Association for Computing Machinery, New York, NY, USA, 2623–2631. https://doi.org/10.1145/3292500.3330701

[6] https://github.com/facebookresearch/fvcore