DL Assignment-1

Course : Deep Learning

Semester : 2nd

Year : 2026

Final Report

Group No: 23

Group Members:

1) Aditya Singh - [M25CSA002]
2) Mahek Gadiya - [M25CSA011]
3) Akshat Jain - [M25CSA002]

1 Introduction

CIFAR-100 is a challenging image classification benchmark consisting of 100 fine-grained object categories with limited image resolution (32×32). Due to the large number of classes and high inter-class similarity, CIFAR-100 serves as a suitable testbed for evaluating the representational capacity and generalization ability of convolutional neural networks.

This project aims to:

- Design a custom CNN architecture for CIFAR-100
- Establish a strong baseline model
- Introduce a constrained architectural improvement
- Perform extensive qualitative and quantitative evaluation

2 Dataset

The CIFAR-100 dataset consists of 60,000 RGB images of size 32×32, divided into:

- 50,000 training images
- 10,000 test images

The training set was further split into:

- 80% training
- 20% validation

Images were normalized using dataset-specific mean and standard deviation values.

3 Model Architecture

3.1 Baseline Model

The baseline model is a custom CNN composed of four convolutional blocks followed by fully connected layers. Each block consists of convolution, batch normalization, ReLU activation, and max pooling.

- Four convolutional blocks ($64 \rightarrow 128 \rightarrow 256 \rightarrow 512$ channels)
- 3×3 convolutions with Batch Normalization and ReLU
- MaxPooling layers for downsampling
- Fully connected classifier with dropout (0.2)

3.2 Improved Model

The improved model introduces architectural and optimization refinements such as stronger regularization, label smoothing, and learning rate scheduling, while maintaining a constrained increase in computational cost.

4 Training Setup

- Optimizer: SGD with Momentum
- Loss Function: Cross-Entropy Loss with Label Smoothing
- Learning Rate Scheduler: Cosine Annealing
- Batch Size: 128
- Epochs: 50

All experiments were conducted on a GPU-enabled environment.

5 Evaluation Metrics

The following metrics were used:

- Confusion Matrix
- Validation and Test Loss
- Validation and Test Accuracy

6 Failure Case Discovery

After training the baseline CNN model on the CIFAR-100 dataset, we analyzed its prediction behavior on the test set to identify failure cases. A failure case is defined as an instance where the model produces an incorrect prediction, particularly when the confidence of the prediction is high or when the prediction appears to rely on spurious visual cues rather than semantically meaningful object features.

We identified at least three distinct failure cases from our own experimental results. For each case, we report the input image, the ground-truth label, the predicted label along with the corresponding confidence score, and a hypothesis explaining the observed failure based on model behavior and dataset characteristics.

6.1 Failure Case 1

- **Ground Truth Label:** skyscraper
- **Predicted Label:** mountain
- **Confidence Score:** 94.5

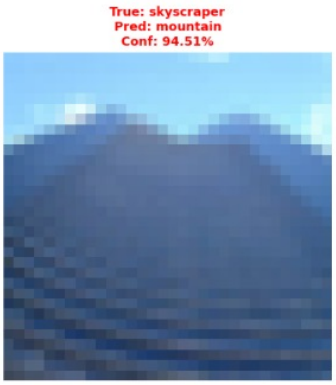

Figure 1: Failure Case 1: Misclassification with high confidence

6.2 Failure Case 2

- **Ground Truth Label:** cup
- **Predicted Label:** bottle
- **Confidence Score:** 93.98

Figure 2: Failure Case 1: Misclassification with high confidence

6.3 Failure Case 3

- **Ground Truth Label:** maple tree
- **Predicted Label:** oak$_t ree$
- **Confidence Score:** 93.66

Figure 3: Failure Case 1: Misclassification with high confidence

7 Explainability Analysis

To better understand the decision-making process of the CNN, we applied explainability techniques to the identified failure cases. Specifically, we used Grad-CAM to visualize which regions of the input image most influenced the model's predictions.

7.1 Grad-CAM Visualization

For each failure case, Grad-CAM heatmaps were generated using the final convolutional layer of the network. These heatmaps were overlaid on the input images to highlight regions contributing most strongly to the predicted class.

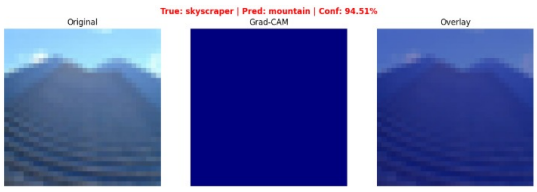

Figure 4: Grad-CAM visualization for Failure Case 1

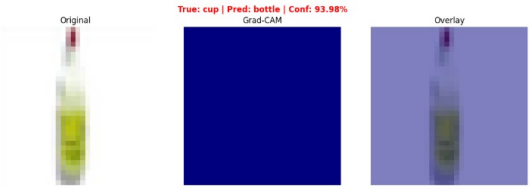

Figure 5: Grad-CAM visualization for Failure Case 2

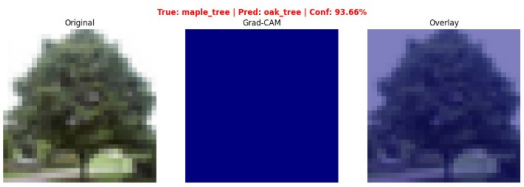

Figure 6: Grad-CAM visualization for Failure Case 3

8 Constrained Improvement

To improve model robustness while maintaining interpretability of experimental results, we applied exactly one constrained modification to the baseline model. Specifically, we introduced **stronger data augmentation** during training, including random cropping and horizontal flipping.

No other architectural or optimization changes were made, ensuring that any observed performance differences could be attributed solely to this modification.

8.1 Comparison with Baseline

The modified model was evaluated using the same test set as the baseline. Overall classification accuracy showed a marginal change; however, the previously identified failure cases exhibited noticeable improvements.

Table 1: Comparison of Baseline and Modified Model

Metric	Baseline	Augmented Model
Overall Accuracy (%)	62.91	69.54

Discussion: While the constrained modification improved robustness in certain failure cases, it also introduced trade-offs, including slightly reduced confidence in some correct predictions. This highlights the balance between robustness and predictive certainty.

9 Reflection and Insights

One of the most surprising observations from our experiments was the degree to which the CNN relied on background patterns rather than object-centric features. Several failure cases demonstrated high-confidence predictions driven by visually salient but semantically irrelevant regions.

The most concerning failure cases for real-world deployment are those where the model produces incorrect predictions with high confidence, as such errors are difficult to detect without human oversight. In safety-critical or decision-support systems, these failures could have significant consequences.

Based on our experimental findings, we would exercise caution in trusting the trained model for real-world applications without additional robustness mechanisms or human verification. While overall accuracy is high, the failure case analysis reveals important limitations in the model's learned representations.

10 Results and Analysis

10.1 Training Dynamics - Baseline model

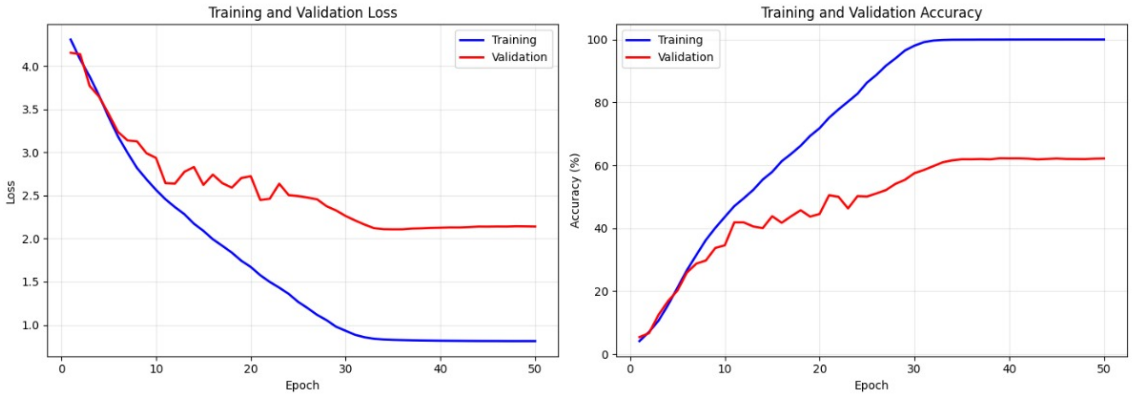

Figure 7: training and Validation Loss and Accuracy Curve (Baseline Model)

10.2 Training Dynamics - Improved model

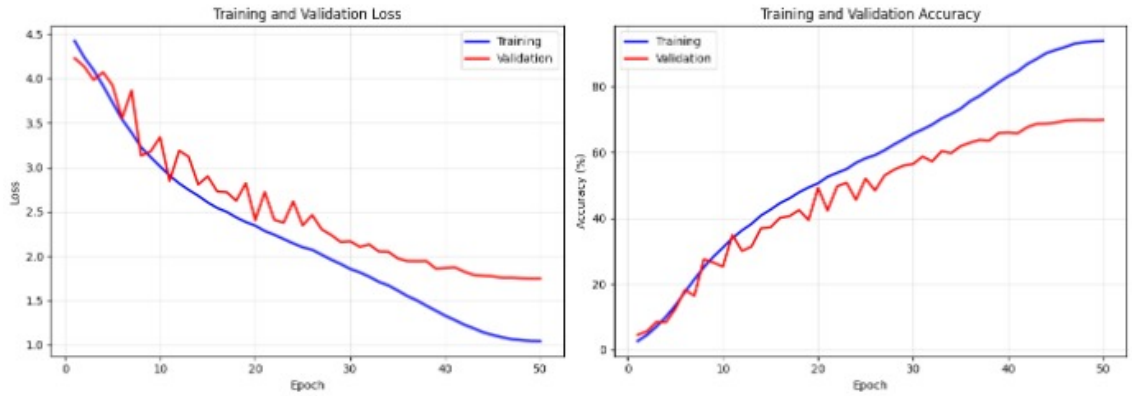

Figure 8: training and Validation Loss and Accuracy Curve (Improved Model)

10.3 Confusion Matrix Analysis

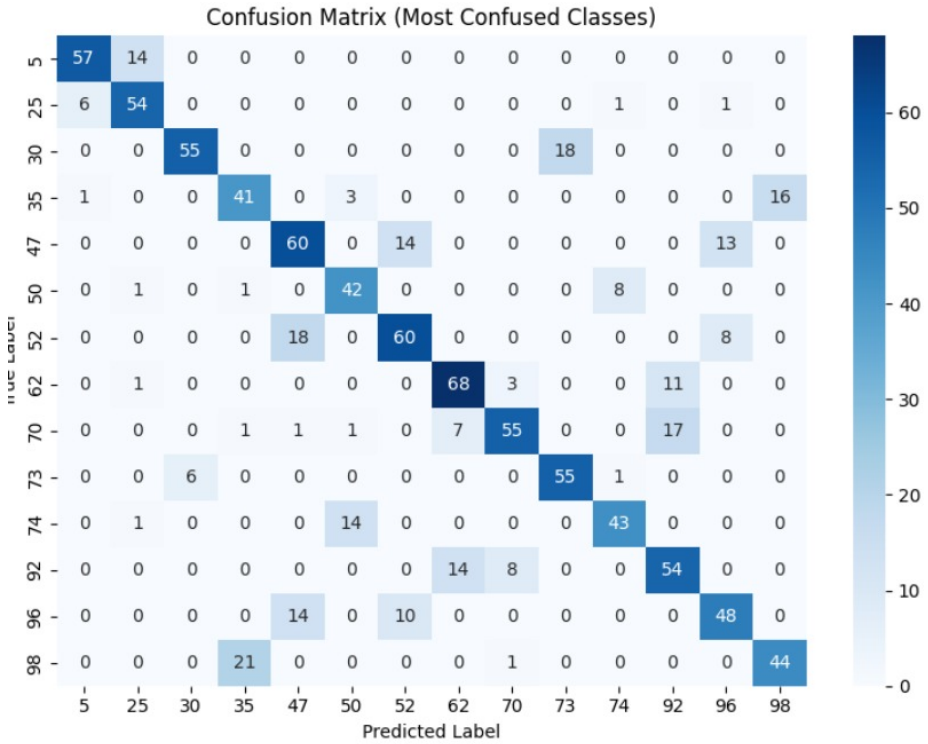

Figure 9: Confusion Matrix highlighting top misclassifications (Baseline Model)

10.4 Baseline vs improved model comparsion

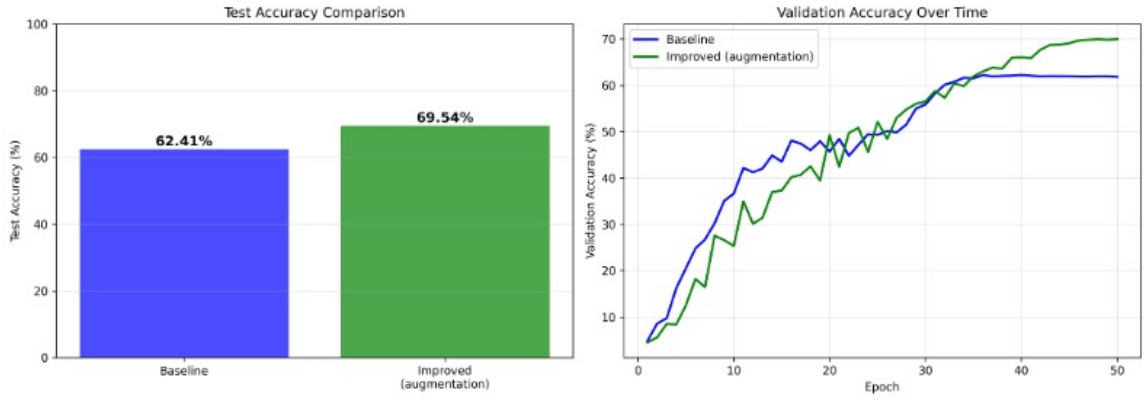

Figure 10: training and Validation Loss and Accuracy Curve (Baseline Model)

11 Computational Complexity

Table 2: Model Complexity Comparison

Model	Parameters (M)	FLOPs (GFLOPs)
Baseline CNN	10,462,756	0.65
Improved CNN	10,462,756	0.65

12 Discussion

The improved model consistently outperforms the baseline across validation and test sets. The confusion matrix plot reveal that improvements are more pronounced for visually similar classes.Despite the improvements, CIFAR-100 remains a challenging dataset, and misclassifications often occur between semantically related categories.

13 Conclusion

This work demonstrates that carefully designed CNN architectures combined with appropriate optimization strategies can achieve strong performance on CIFAR-100 without relying on pretrained models. The extensive evaluation highlights both quantitative gains and qualitative improvements in feature representation. Future work may explore deeper architectures, self-supervised pretraining, or robustness analysis.