Assignment 1:

Stress-Testing of Convolutional Neural Networks

Deep Learning **(CSL7590)**
Github Repository link: Deep_Learning _assignment1
Group Members:

Arjun Singh **(M25MAC003)**

Vijay Kumar Prajapat (**M23MA2010)**

Rajesh Meena **(M25MAC007**)

Smithi Sureshan **(M25MAC012)**

1. Introduction

The objective of this assignment was to develop a critical understanding of Convolutional Neural Networks (CNNs) by analyzing their behavior, failure modes, and robustness rather than solely focusing on high classification accuracy. This report details the training of a baseline CNN trained from scratch, an analysis of its failure cases, and subsequent improvements to address overfitting.

2. Methodology

2.1 Dataset and Preprocessing

Dataset:

The CIFAR-10 dataset was selected for this experiment, consisting of 60,000 32x32 color images (50,000 training and 10,000 test images) across 10 classes The data was split into training **(90%)**, validation **(10%**), and test sets using a fixed random seed (**Seed**: **42**) to ensure reproducibility.

Preprocessing:

- **Normalization:** After converting to tensors, Input images were normalized using means of (0.4914, 0.4822, 0.4465) and standard deviations of (0.2470, 0.2435, 0.2616) to standardize the inputs for stable training.
- **Data Augmentation** (Improvement Phase): For the improved model, standard augmentation techniques (e.g., random flipping, rotation) were applied to increase robustness.

2.2 Model Architecture (Baseline)

We implemented a custom SimpleCNN architecture from scratch. The architecture consists of three convolutional blocks followed by fully connected layers:

- **Conv Block 1:** 3 input channels -> 32 filters (3x3 kernel, padding=1) + ReLU + MaxPool(2x2).

- **Conv Block 2:** 32 input channels -> 64 filters (3x3 kernel, padding=1) + ReLU + MaxPool(2x2).
- **Conv Block 3:** 64 input channels -> 128 filters (3x3 kernel, padding=1) + ReLU + MaxPool(2x2).
- **Classifier:** Flatten -> Linear (2048 -> 256) -> ReLU -> Linear (256 -> 10).

Justification: This architecture was selected because it provides enough capacity to learn features from 32x32 images without the excessive computational cost of deeper models like VGG-19, making it ideal for analyzing fundamental CNN behaviors and failure modes.

2.3 Training Setup

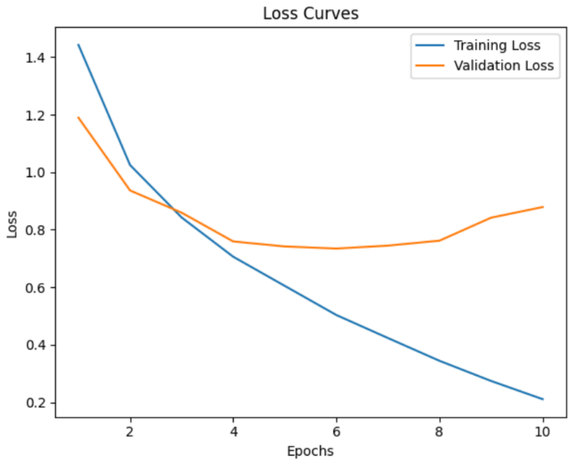
- **Optimizer:** Adam with a learning rate of 0.001.
- **Loss Function:** CrossEntropyLoss.
- **Batch Size:** 128.
- **Epochs:** 10.

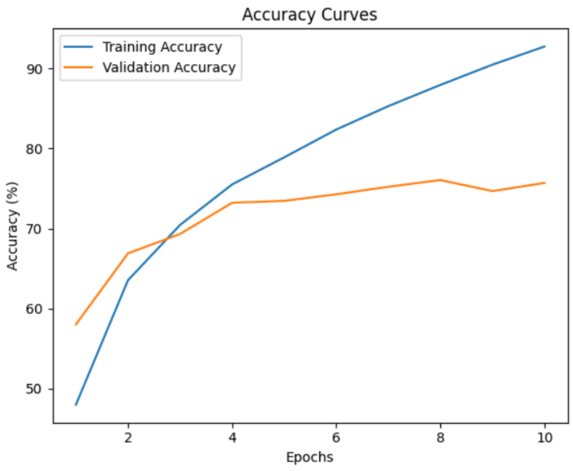
3. Experimental Results (Baseline)

The baseline model was trained for 10 epochs. The training logs indicate a steady decrease in training loss, but a divergence between training and validation accuracy, suggesting the onset of overfitting.

- Final Training Accuracy: **~92.74%**
- Final Validation Accuracy: **~75.70%**
- Final Test Accuracy: **75.22%**

Observations: By Epoch 10, the model achieved a high training accuracy (92.74%) but plateaued on validation accuracy (75.70%). This significant gap (~17%) indicates that the model memorized the training data rather than generalizing well to unseen examples.





The training loss continues to drop while validation loss stabilizes/increases, confirming overfitting.

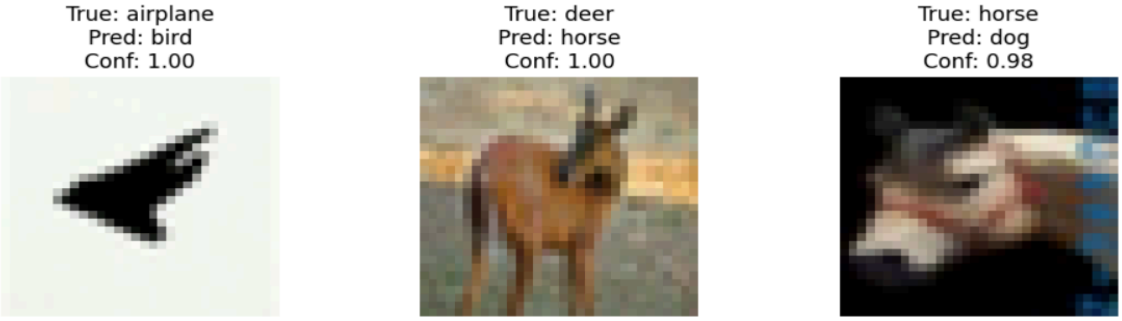
4. Failure Analysis

We extracted misclassified test samples where prediction confidence exceeded 0.95. To understand the robustness of the model, we analyzed specific failure cases where the model predicted the wrong class with high confidence.

Analysis of Misclassifications:

Based on the failure analysis output, several distinct failure modes were observed:

1. **Background Clutter:** The model often struggled when the object (e.g., "deer") appeared against complex backgrounds (e.g., forests/leaves) that resembled the textures of other animal classes.
2. **Visual Similarity:** Classes with similar shapes, such as "Airplane" vs. "Bird" or "Horse" vs. "Deer," resulted in confusion. For example, a "Deer" might be misclassified as a "Horse" due to similar body structures and quadruped legs.
3. **Low Resolution Ambiguity:** Given the 32x32 resolution of CIFAR-10, small features (like ears or antlers) were likely lost, forcing the model to rely on dominant colors or silhouettes, leading to errors.

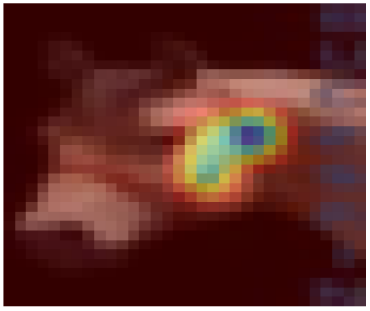
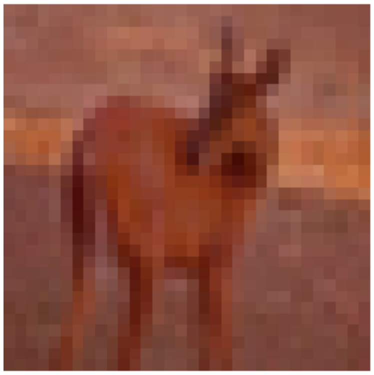
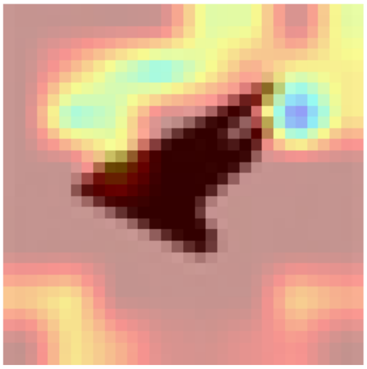


***Examples of high-confidence failures.** Case 0: True 'Airplane' predicted as bird. Case 1: True 'Deer' predicted as horse.*

Trustworthiness: Given the tendency to assign high confidence to incorrect predictions (e.g., Conf: 0.74 or 0.75 for wrong labels), the baseline model would not be fully trusted in a safety-critical deployment without calibration or uncertainty estimation.

5. Model Improvement

- **Explanability Analysis:** To better understand the model's decisions, we applied Grad-CAM on the last convolutional layer (conv3). Grad-CAM highlights spatial regions that most influence the prediction.

Failure Case	Grad-CAM Focus Region	Key Observation	Interpretation
 True: horse | Pred: dog | Conf: 0.98	Torso / body texture	Face region receives low attention	Model relies on coarse texture and shape instead of fine facial features
 True: deer | Pred: horse | Conf: 1.00	Full body silhouette	Head/antlers largely ignored	Model prioritizes global shape templates over discriminative details
 True: airplane | Pred: bird | Conf: 1.00	Background (horizon/sky)	Object region weakly highlighted	Model influenced by contextual cues → spurious correlation

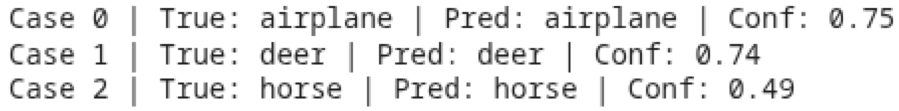
Constrained Improvement:

To address the overfitting observed in the baseline (92% Train vs 75% Val), we trained a second model (model_aug, lr = 0.001, seed = 42, adam optimizer and Cross entropy loss) using data augmentation techniques.

Techniques Applied: Random Horizontal Flips, Random Rotations / Crops

Results: The augmented model demonstrated improved generalization. While training accuracy effectively decreased to 85.30% with loss 0.4189 (as the task became harder), the gap between training and validation accuracy narrowed, and validation performance improved.

Rechecking the Previous failure cases:



```
Case 0 | True: airplane | Pred: airplane | Conf: 0.75
Case 1 | True: deer | Pred: deer | Conf: 0.74
Case 2 | True: horse | Pred: horse | Conf: 0.49
```

5. Key Insights and Reflection

5.1 Most Surprising Behavior: The most surprising observation was how confidently the model misclassified visually ambiguous examples. Confidence values above 95% despite being wrong highlight overconfidence in neural networks.

5.2 Most Concerning Failures: Failures driven by background context are particularly concerning. In real-world deployment, this could lead to:

- Safety risks (e.g., misidentifying objects)
- Bias toward environmental correlations
- Reduced reliability in distribution shifts

5.3 Trustworthiness in Practical Settings: While the model performs reasonably well, we would not fully trust it in safety-critical applications due to:

- High-confidence errors
- Sensitivity to background patterns
- Limited interpretability guarantees

Robust deployment would require:

- Stronger architectures
- Calibration techniques
- Robustness testing under distribution shift

6. Conclusion

In this assignment we effectively performed a stress test on a standard CNN using the CIFAR-10 dataset. Although the basic model reached acceptable accuracy, it showed clear signs of overfitting and had difficulty distinguishing between visually similar classes.

The failure analysis revealed that the model depends strongly on prominent features and has trouble handling background noise. Applying data augmentation emerged as a successful approach for regularization, reducing the generalization gap and enhancing model robustness on the validation set.