Final Semester Report

CIFAR-10 Convolutional Neural Network (CNN) Comparison Project

Authors: Tamuka Manjemu & Tavonga Dutuma
Institution: Quinnipiac University
Course: Neural Networks
Instructor: Prof. Soumyashree Sahoo

---

1. Introduction

This project investigates and compares three Convolutional Neural Network (CNN) architectures on the CIFAR-10 dataset, a standard benchmark for image classification tasks. The goal is to understand how model depth, normalization techniques, and data augmentation influence classification accuracy, generalization, and computational efficiency.

The project demonstrates full-cycle experimentation, including architecture design, training, evaluation, and results interpretation.

---

2. Research Question

What CNN architecture provides the optimal balance of accuracy, efficiency, and generalization for CIFAR-10 image classification?

---

3. Dataset Overview

- **Dataset:** CIFAR-10
- **Images:** 60,000 color images (32×32)
- **Classes:** 10 (animals and vehicles)
- **Train/Test Split:** 50,000 train, 10,000 test
- **Challenges:**
  - Low resolution
  - High inter-class similarity
  - Noisy samples

---

4. Model Architectures

Three models were developed for comparison:

4.1 Baseline CNN

Architecture:
Input → [Conv(32) → Conv(32) → MaxPool → Dropout] ×2 → Flatten → Dense(512) → Softmax
Parameters: ~1.2M
Purpose: Provide a lightweight baseline.

4.2 CNN with Data Augmentation

Architecture:
Input → RandomFlip → RandomRotation → Convolutional Blocks → Dense Classifier
Parameters: ~1.8M
Purpose: Study generalization using augmentation.

4.3 Deeper CNN

Architecture:
Input → [Conv → BatchNorm → Conv → MaxPool → Dropout] ×3 → GlobalAvgPool → Dense → Softmax
Parameters: ~2.5M
Purpose: Test effects of increased depth and normalization.

---

5. Experimental Setup

- **Framework:** TensorFlow/Keras
- **Optimizer:** Adam
- **Loss Function:** Categorical Cross-Entropy
- **Epochs:** 20–30
- **Metrics:** Training accuracy, validation accuracy, test accuracy
- **Hardware:** CPU or GPU (GPU recommended)

---

6. Results Summary

Model	Test Accuracy	Key Features	Insight
Deeper CNN	**77.62%**	BatchNorm, GAP, 3 conv blocks	Best overall performance
Baseline CNN	**77.02%**	Simple structure + dropout	Surprisingly strong and efficient
CNN with Augmentation	**71.10%**	RandomFlip/Rotation	Over-augmentation hurt accuracy

---

7. Analysis & Interpretation

7.1 What Worked Best

- **Simple models can perform very well.**
  The baseline model nearly matched the deeper architecture.
- **Depth + batch normalization improved stability.**
  Better convergence and slightly higher accuracy.
- **Global Average Pooling** reduced parameters while maintaining strong performance.

7.2 Key Challenges & Lessons Learned

- **Overly aggressive augmentation worsened accuracy.**
- **Small accuracy gains from deeper models** on CIFAR-10.
- **Animal classes exhibited the highest confusion**, due to shared features.
- **Some architectures showed validation–test discrepancies**, indicating sensitivity to distribution shifts.

---

8. Educational Value & Skills Developed

This project provided hands-on experience with:

- Designing CNN architectures
- Implementing and training deep learning models
- Applying normalization and regularization
- Performing systematic model comparisons
- Visualizing loss/accuracy trends
- Conducting classification error analysis

These skills align with core learning outcomes of the Neural Networks course.

---

9. Installation & Reproducibility

```
git clone https://github.com/yourusername/cifar10-cnn-comparison.git
cd cifar10-cnn-comparison
pip install -r requirements.txt
jupyter notebook main.ipynb
```

A Google Colab badge is included in the repository for GPU-based execution.

---

10. References

- Krizhevsky, A. (2009). *Learning Multiple Layers of Features from Tiny Images.*
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep Residual Learning for Image Recognition.*
- Ioffe, S. & Szegedy, C. (2015). *Batch Normalization: Accelerating Deep Network Training.*
- Simard, P., Steinkraus, D., & Platt, J. (2003). *Best Practices for Convolutional Neural Networks.*