Caltech-256 Image Classification — Project Report

Author: Shreyas Rajeev Patil

Built Using: Custom Convolutional Neural Network (CNN)

Dataset: Caltech-256

Final Test Accuracy: 81.21%

1. Objective

The project aimed to develop a complete end-to-end image classification system on the Caltech-256 dataset using a fully custom-designed CNN. Steps included dataset preparation, transformations, model design, training, evaluation, and model saving.

2. Dataset Preparation

The dataset contains 30,607 images across 257 categories. After extraction, it was split into 70% training, 15% validation, and 15% test sets, ensuring clean folder structure and correct class distribution.

3. Data Transformations

Training data was resized, normalized, and augmented with horizontal flipping. Validation and test data were resized and normalized without augmentation.

4. Data Loading

Batch size was set to 32 with shuffling enabled for training. Final counts: 868 training batches, 259 validation batches, and 279 test batches.

5. Custom CNN Architecture

A deep CNN was built with 4 convolution blocks (64→128→256→512 filters), each containing two convolution layers with ReLU and MaxPooling. Fully connected layers consist of 4096→4096→257 units with dropout for regularization. Label smoothing and learning rate scheduling were used.

6. Training Process

Training was performed for 20 epochs in two phases. Loss decreased steadily, and validation accuracy improved consistently with stable learning and no overfitting.

7. Results

The final test accuracy achieved was 81.21%, which is strong for a dataset with 257 diverse object categories.

8. Confusion Matrix Analysis

The confusion matrix showed strong diagonal dominance, indicating accurate predictions across most categories with limited confusion in visually similar classes.

9. Training & Validation Curves

The loss curves showed stable downward trends, and accuracy increased smoothly from ~8% to over 80%, demonstrating efficient learning.

10. Sample Predictions

The model produced correct predictions for most test samples, handling variations in lighting, angle, and background effectively.

11. Model Saving

The trained model was saved as final_model.pth for future inference or fine-tuning.

12. Conclusion

The project demonstrates a robust end-to-end implementation of a custom CNN for multi-class image classification, achieving high performance on a complex dataset with strong generalization.