# CIFAR-10 Image Classification with Deep Learning

## Project Overview

This project implements a Convolutional Neural Network (CNN) to classify images from the CIFAR-10 dataset, which contains 60,000 32x32 RGB images across 10 classes (e.g., airplane, cat, dog). The model leverages advanced techniques like data augmentation, dropout, batch normalization, and early stopping to achieve robust performance (~75-85% test accuracy). The project generates three visualizations: training/validation accuracy and loss curves, sample predictions, and a confusion matrix to analyze class performance.

### Objectives

* Build and train a CNN for image classification.
* Apply data augmentation and regularization to improve generalization.
* Visualize model performance through plots and metrics.

### Unique Features

* **Data Augmentation**: Random rotations, flips, shifts, and zooms to enhance model robustness.
* **Regularization**: Dropout (25% and 50%) and batch normalization to prevent overfitting.
* **Early Stopping**: Optimizes training by halting when validation loss plateaus.
* **Confusion Matrix**: Identifies class-specific performance and misclassifications.

## Prerequisites

* **Python**: Version 3.7–3.10
* **Libraries**:
  + TensorFlow (>=2.10)
  + Matplotlib
  + NumPy
  + Scikit-learn
  + Seaborn
* **Hardware**: Runs on CPU (~10-15 minutes) or GPU (faster, e.g., Google Colab).

## Installation

1. **Clone or Download**: Obtain the project files (e.g., cifar10\_cnn\_extended\_corrected.py).
2. **Set Up a Virtual Environment** (recommended):

python -m venv env

source env/bin/activate # On Windows: env\Scripts\activate

1. **Install Dependencies**:

pip install tensorflow matplotlib numpy scikit-learn seaborn

## Usage

1. **Save the Code**: Ensure cifar10\_cnn\_extended\_corrected.py is in your working directory.

**Run the Script**:

python cifar10\_cnn\_extended\_corrected.py

1. **Outputs**:
   * **Console**: Displays test accuracy (~75-85%) and plot save confirmations.
   * **Files** (saved in the working directory):
     + training\_plots.png: Training/validation accuracy and loss curves.
     + sample\_predictions.png: 10 test images with predicted/true labels.
     + confusion\_matrix.png: Heatmap of true vs. predicted classes.

## Project Structure

* **cifar10\_cnn\_extended\_corrected.py**: Main script containing:
  + Data loading and preprocessing (CIFAR-10 dataset).
  + CNN model with three convolutional blocks, dropout, and batch normalization.
  + Data augmentation (random transformations).
  + Training with early stopping.
  + Visualization of training metrics, sample predictions, and confusion matrix.
* **Output Files**:
  + training\_plots.png
  + sample\_predictions.png
  + confusion\_matrix.png

## Expected Results

* **Test Accuracy**: Approximately 75-85% after ~15-25 epochs (early stopping may halt earlier).
* **Training Plots**: Show increasing accuracy and decreasing loss, with validation metrics stabilizing.
* **Sample Predictions**: Displays 10 test images; most should be correctly classified, though some (e.g., cat vs. dog) may show errors.
* **Confusion Matrix**: Highlights strong performance (high diagonal values) and potential class confusions (e.g., cat/dog).

## Troubleshooting

* **Syntax Errors**: Ensure all dependencies are installed and Python version is compatible.
* **Memory Issues**: If running on a low-RAM system, try reducing batch\_size (e.g., to 32) in the script.
* **No Plots Generated**: Verify Matplotlib and Seaborn are installed; check the working directory for output files.
* **TensorFlow Errors**: Use TensorFlow 2.10+ and ensure GPU drivers are updated if using a GPU.

## Extending the Project

* **Hyperparameter Tuning**: Adjust learning rate, batch size, or epochs.
* **Advanced Models**: Use pre-trained models (e.g., ResNet50) for transfer learning.
* **More Augmentation**: Add color jittering or cutout.
* **Interactive Demo**: Build a web interface with Flask/Streamlit to upload and classify images.

## License

This project is for educational purposes and can be freely used or modified for academic assignments.

## Acknowledgments

* CIFAR-10 dataset: Alex Krizhevsky, University of Toronto.
* TensorFlow/Keras for deep learning framework.
* Matplotlib and Seaborn for visualizations.