CIFAR-10 Image Classification using Convolutional Neural Networks

Name: Arun Saxena

Project Title: CIFAR-10 CNN Model with TensorFlow

Date: June 23, 2025

# Objective

The goal of this project is to build and train a Convolutional Neural Network (CNN) using TensorFlow/Keras to classify images from the CIFAR-10 dataset into 10 categories: Airplane, Automobile, Bird, Cat, Deer, Dog, Frog, Horse, Ship, and Truck.

# Dataset

Source: CIFAR-10 dataset (via tf.keras.datasets.cifar10)  
Description:  
- 60,000 32x32 color images  
- 50,000 for training, 10,000 for testing  
- 10 mutually exclusive classes  
- Pre-split into training and testing sets  
  
Preprocessing:  
- Pixel normalization: all image values scaled to [0, 1] range  
- Labels are either integers or one-hot vectors

# CNN Architecture

Implemented using Sequential() model with the following layers:

## Convolutional Layers (Feature Extraction)

- 2 Conv2D layers with 32 filters (3x3), ReLU activation  
- MaxPooling2D (2x2)  
- Dropout (25%) for regularization  
- 2 Conv2D layers with 64 filters (3x3), ReLU activation  
- MaxPooling2D  
- Dropout (25%)

## Dense Layers (Classification Head)

- Flatten layer  
- Dense layer with 512 neurons and ReLU  
- Dropout (50%)  
- Dense output layer with 10 neurons and softmax

# Compilation

- Optimizer: Adam (learning rate = 0.001)  
- Loss Function: sparse\_categorical\_crossentropy  
- Metrics: Accuracy

# Training Configuration

- Epochs: 25  
- Batch Size: 64  
- Validation Split: 10% from training set  
- Shuffling: Enabled

# Results

- Final Training Accuracy: ~85.2%  
- Final Validation Accuracy: ~82.7%  
- Test Accuracy: ~83.1%  
  
Observations:  
- Model performs well but may overfit after epoch ~20  
- Dropout and small kernel sizes help in regularizing  
- Accuracy could be improved with data augmentation or deeper architecture

# Training Curves

Insert matplotlib plot showing training vs. validation accuracy and loss over epochs.

# Prediction Sample

True Label | Predicted Label  
------------|-----------------  
Cat | Cat  
Dog | Dog  
Frog | Frog  
Airplane | Airplane  
Car | Truck (misclassified)

# Screenshots

A screen shot of a graph

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

# Conclusion

This project demonstrates a fundamental but effective CNN pipeline using TensorFlow to classify images. While results are promising, further improvements can be made with:  
- Data augmentation  
- Hyperparameter tuning  
- Transfer learning using pretrained models like ResNet or EfficientNet