Project 2 - CIFAR-10 Image Classification

Project Overview

This project involves building an image classification system using the CIFAR-10 dataset. Two models were developed and compared: - A **Custom Convolutional Neural Network (CNN)** built from scratch - A **Transfer Learning Model** using MobileNetV2 pretrained on ImageNet

Both models are evaluated, visualized, and analyzed according to full deep learning best practices.

Data Preprocessing

- **Dataset:** CIFAR-10 (60,000 32x32 color images, 10 classes)
- **Normalization:** Pixel values scaled to [0, 1]
- **Resizing:** Images resized to 160x160 for MobileNetV2 compatibility
- **Augmentation:** (Not applied, optional future improvement)
- **Visualization:** 10 random CIFAR-10 images displayed with their labels.

Model Architectures

1. Custom CNN Architecture

- **Layers:**
  - Conv2D (32 filters) + MaxPooling
  - Conv2D (64 filters) + MaxPooling
  - Flatten
  - Dense(128) + Dropout(0.3)
  - Output Dense(10) with Softmax

2. Transfer Learning - MobileNetV2

- **Base:** MobileNetV2 pretrained on ImageNet
- **Custom Top Layers:**
  - GlobalAveragePooling
  - Dense(128) + Dropout(0.3)
  - Output Dense(10) Softmax layer
- **Training Strategy:**
  - First, freeze base layers and train head only
  - Then, unfreeze top 50 layers and fine-tune

Model Training Details

- **Optimizer:** Adam
- **Batch Size:** 32 (for MobileNetV2), 64 (for Custom CNN)
- **Epochs:**
  - 10 epochs for initial MobileNetV2 training
  - 10 additional epochs for fine-tuning
  - 20 epochs for Custom CNN (EarlyStopping applied)
- **Callbacks:** EarlyStopping used for fine-tuning and CNN

Results and Analysis

Custom CNN

- **Test Accuracy:** ~70%-75%
- **Observations:**
  - Basic CNN performs decently without augmentation.
  - Potential improvement with data augmentation.

Transfer Learning Model

- **Test Accuracy:** ~90%-91%
- **Observations:**
  - Transfer learning greatly boosts performance.
  - Fine-tuning slightly improves MobileNetV2 further.

Confusion Matrix

- Generated for Custom CNN.
- Most confusion between similar categories (e.g., cat/dog, truck/automobile).

Precision, Recall, F1-Score

- Detailed classification report produced for Custom CNN.

Best Model and Why

- **Best Model:** MobileNetV2 with Fine-Tuning
- **Reason:** Achieves higher accuracy (~90%) compared to Custom CNN (~70%), faster convergence, and robustness.

Insights

- Transfer learning is extremely powerful even with small input images.

- Fine-tuning is crucial to extract maximum performance.
- Proper preprocessing (resizing, normalization) and EarlyStopping significantly help model stability.

Future Work

- Add data augmentation to Custom CNN.
- Try other pre-trained models (ResNet50, EfficientNet).
- Deploy best model via TensorFlow Serving.

Files Included

- `Project2_G2_complete.ipynb`: Main Jupyter Notebook
- `REPORT.pdf`: Full report
- `PPT Presentation`: Slides summarizing the project
- `requirements.txt`: List of necessary packages
- `Gradio or Flask App`: For model deployment (separate if needed)

End of Report