Project Report: Object Recognition using ResNet50

1. Project Overview

This project focuses on the task of object recognition, a core problem in computer vision. The goal is to classify images from the CIFAR-10 dataset into one of 10 categories. The project compares two distinct approaches: a basic, fully-connected neural network and a more sophisticated model based on the pre-trained ResNet50 architecture. The notebook provides a complete workflow, from data preparation to model training and evaluation.

The CIFAR-10 dataset consists of 60,000 32×32 color images in 10 classes, with 6,000 images per class. The 10 classes are: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.

2. Data Preparation and Preprocessing

The project begins by setting up the environment and downloading the dataset from Kaggle. The raw data is a compressed archive containing 50,000 training images and 10,000 test images. The key data preprocessing steps include:

- Extraction: The compressed dataset files (cifar-10.zip, train.7z, test.7z) are extracted.
- Labeling: The image files, which are named by their ID, are paired with their corresponding labels from trainLabels.csv.
- Data Conversion: The image data is read using the PIL and OpenCV libraries, then converted into NumPy arrays. Each image is a 32×32×3 array of pixel values.
- Splitting: The 50,000 training images and their labels are split into training and testing sets, with 80% (40,000 images) for training and 20% (10,000 images) for testing. This is a crucial step for evaluating the model's performance on unseen data.
- Scaling: The pixel values of the images are normalized

from the range [0,255] to a [0,1] scale. This is a standard practice in machine learning that helps the model converge faster and perform better.

3. Model 1: Simple Neural Network

To establish a baseline, a simple sequential neural network is built using TensorFlow/Keras. This model architecture consists of:

- Flatten Layer: The 32×32×3 image is flattened into a one-dimensional vector of size 3072.
- Dense Layer: A hidden layer with 64 neurons and a ReLU activation function.
- Output Layer: A final dense layer with 10 neurons (one for each class) and a Softmax activation function to produce probability scores.

The model is compiled with the Adam optimizer and a sparse_categorical_crossentropy loss function. After training for 10 epochs, the model achieves a test accuracy of approximately 25%, demonstrating that this simple approach is not sufficient for the complexity of the CIFAR-10 dataset.

4. Model 2: ResNet50 - A Deep Learning Approach

The core of this project is the implementation of a Convolutional Neural Network (CNN) using the ResNet50 architecture.

4.1. What is ResNet50?

ResNet, short for Residual Network, is a groundbreaking CNN architecture introduced by Kaiming He et al. in 2015. ResNet50 is a specific variant of this architecture with 50 layers. The key innovation of ResNet is the use of "skip connections" or "residual blocks."

- The Vanishing Gradient Problem: As neural networks get deeper, a common problem known as the "vanishing gradient" occurs. During backpropagation, the gradients become

smaller as they are propagated backward through the layers, making the weights of the initial layers difficult to train.

- Residual Blocks: ResNet addresses this by introducing a direct connection, or "shortcut," that skips one or more layers. In a residual block, the output of a layer is added to the input of that same layer. This allows the network to learn a "residual mapping" rather than the entire function. Essentially, it helps the network learn F(x) + x instead of just F(x). This simple modification enables the training of much deeper networks without performance degradation.

The ResNet50 architecture consists of several convolutional layers, pooling layers, and these residual blocks. It was originally trained on the ImageNet dataset, which contains millions of images across 1000 classes.

4.2. Implementation in the Project

The project leverages a pre-trained ResNet50 model from Keras to benefit from the rich features it has already learned. The implementation details are as follows:

- Convolutional Base: The pre-trained ResNet50 model, with weights from ImageNet, is loaded. The include_top=False argument is used to exclude the final classification layers, allowing the project to add its own custom layers for the CIFAR-10 classification task.
- Image Resizing: The CIFAR-10 images are only 32×32 pixels, which is much smaller than the standard input size for ResNet50 (typically 224×224 or 256×256). The code addresses this by up-sampling the images several times using UpSampling2D layers to match the expected input shape of the ResNet50 base.
- Transfer Learning: The pre-trained weights from the ImageNet dataset are kept, allowing the model to act as a powerful feature extractor.

- Custom Classification Head: A new classification head is added on top of the ResNet50 convolutional base. This consists of:
  - Flatten Layer: To convert the 3D output of ResNet50 into a 1D vector.
  - Batch Normalization & Dropout: These layers are added to regularize the model and prevent overfitting.
  - Dense Layers: A series of fully-connected layers with ReLU activation to learn the new classification task.
  - Output Layer: A final dense layer with 10 neurons and a Softmax activation for the 10 CIFAR-10 classes.
- Fine-Tuning: The model is compiled with a very small learning rate (2e-5) using the RMSprop optimizer. This is a common practice in transfer learning, allowing the new custom layers to learn while minimally adjusting the pre-trained weights.
- Training & Evaluation: The model is trained for 10 epochs. The results show a remarkable improvement in accuracy. The final test accuracy achieved is approximately 93.6%, a significant leap from the 25% of the simple neural network.

5. Conclusion

The project successfully demonstrates the power of transfer learning and the ResNet50 architecture for object recognition. By leveraging a pre-trained model as a feature extractor, the project achieved a high level of accuracy on a new dataset with minimal training, far surpassing the performance of a simple neural network.