**Image Classification Using CNN**

**Author: Zaryab Ahmad Khan**

Objective

The objective of this project is to build and evaluate different models for image classification using the CIFAR-10 dataset. The focus is on using Convolutional Neural Networks (CNNs) and exploring their performance.

1. Introduction
2. This report details the steps taken to prepare the data, build models, train and evaluate them, and the final results achieved. The CIFAR-10 dataset is used, which consists of 60,000 32x32 color images in 10 different classes. The goal is to classify these images into their respective categories.
3. Data Preparation
4. **Dataset**:
5. CIFAR-10
6. **Loading the Data**:

The dataset is loaded using the TensorFlow library, which provides a convenient way to access the CIFAR-10 dataset. The data is split into training and test sets.

1. **Data Shape**:
   * + - 1. The shapes of the training and test data are printed to understand the structure and dimensions of the dataset. The training set contains 50,000 images, and the test set contains 10,000 images, each of size 32x32 pixels with 3 color channels (RGB).
2. **Reshaping Labels**:
   * + - 1. The labels for the images are reshaped from a 2D array to a 1D array to match the required format for model training.
3. **Class Labels**:
   * + - 1. The class labels are defined to represent the 10 different categories in the CIFAR-10 dataset, which include classes like airplane, automobile, bird, cat, etc.
4. **Data Normalization**:
   * + - 1. The pixel values of the images are normalized by dividing by 255 to scale them to a range of 0 to 1. This helps in faster convergence during training.
5. Model Architectures
6. **Artificial Neural Network (ANN) Model**
   * 1. A simple ANN model was first created as a baseline:
7. **Model Definition**:
   * 1. The ANN model consists of a series of fully connected layers (dense layers). The first layer flattens the input image, and the subsequent layers apply ReLU activation. The final layer uses a softmax activation to output probabilities for the 10 classes.
8. **Model Compilation**:
   * + 1. The model is compiled using the Stochastic Gradient Descent (SGD) optimizer, with sparse categorical crossentropy as the loss function, and accuracy as the evaluation metric.
9. **Model Training**:
   * 1. The model is trained on the training data for 5 epochs. An epoch is one complete pass through the entire training dataset.
10. Training Process
11. The ANN model was trained using:
12. **Optimizer**: SGD (Stochastic Gradient Descent)
13. **Loss Function**: Sparse Categorical Crossentropy
14. **Metrics**: Accuracy
15. **Epochs**: 5
16. Evaluation Metrics
17. The performance of the ANN model was evaluated using a confusion matrix and a classification report.
18. **Predictions**:
    * + 1. The model's predictions on the test set are obtained. Each prediction is the class with the highest probability.
19. **Confusion Matrix and Classification Report**:
    * + 1. The confusion matrix provides a summary of the number of correct and incorrect predictions, while the classification report gives detailed metrics like precision, recall, and F1-score for each class.
20. Results
    * 1. The ANN model provided a basic performance overview with limited accuracy. The confusion matrix and classification report indicated the model's ability to classify certain classes better than others.
21. Conclusion
22. The initial model built was a simple ANN which served as a baseline. Future work will involve:
23. Building more complex CNN architectures to improve accuracy.
24. Exploring and implementing model compression techniques like pruning, quantization, and knowledge distillation.
25. Hyperparameter tuning for optimizing the model's performance.