

Project Group 8: CIFAR-10 Object Recognition

Team Details:

Team Name: Atomax

Members:

1. Kunal Verma
2. Debojit Roy Chowdhury
3. Moulindu Mapa

1. Problem Statement

The objective of this project is to develop a Convolutional Neural Network (CNN) for object recognition using the CIFAR-10 dataset. The task involves classifying input images into one of ten predefined object categories based on visual features learned during training.

The project aims to demonstrate an understanding of deep learning concepts such as convolution, pooling, feature extraction, and classification, rather than achieving state-of-the-art performance. In addition to evaluating the model on the standard test dataset, the trained network is further validated using images obtained from the internet to assess its generalization capability on unseen real-world data.

2. Dataset Description

The dataset used in this project is the CIFAR-10 dataset, a widely used benchmark dataset for image classification tasks. The dataset consists of 60,000 color images of size 32×32 pixels, distributed equally across ten object classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.

The dataset is divided into 50,000 training images and 10,000 testing images. Each image is represented as an RGB image with three color channels. Prior to training, pixel values are normalized to the range $[0, 1]$ to improve numerical stability during optimization. Class labels are converted into one-hot encoded vectors to enable multi-class classification using categorical cross-entropy loss.

The CIFAR-10 dataset is chosen due to its balanced class distribution and suitability for evaluating convolutional neural networks.

3. Methodology

The methodology adopted in this project is based on supervised deep learning using a Convolutional Neural Network (CNN), which is well suited for image classification tasks due to its ability to automatically learn spatial hierarchies of features from raw pixel data. Unlike traditional machine learning approaches that rely on handcrafted features, CNNs learn relevant visual patterns directly through training.

The proposed approach begins with preprocessing the input images through normalization. The normalized images are passed through convolutional layers that extract low-level and progressively higher-level visual features, which are later used for classification.

Max-pooling layers are employed after convolutional layers to reduce the spatial dimensions of feature maps, thereby decreasing computational complexity and providing a degree of translation invariance. The extracted features are subsequently flattened and fed into fully connected layers that perform the final classification task.

The network is trained using the backpropagation algorithm with categorical cross-entropy as the loss function and optimized using the Adam optimizer. Model performance is evaluated on

a held-out test dataset as well as on external images collected from the internet to assess the generalization capability of the trained model on unseen data.

4. Experimental Details

4.1. Architecture Description

The convolutional neural network designed for this project follows a sequential architecture consisting of convolutional, pooling, and fully connected layers. The input to the network is a 32×32 RGB image.

The first convolutional layer applies multiple filters with a small receptive field to capture low-level visual features such as edges and textures. This is followed by a max-pooling layer that reduces the spatial dimensions of the feature maps while retaining the most significant information. A second convolutional layer is then used to extract higher-level features, followed again by max-pooling to further reduce dimensionality.

The resulting feature maps are flattened and passed through a fully connected layer with ReLU activation, enabling the network to learn non-linear combinations of the extracted features. The final output layer consists of ten neurons with a SoftMax activation function, corresponding to the ten classes in the CIFAR-10 dataset.

A detailed summary of the network architecture, including layer types, output shapes, and the number of trainable parameters, is provided using TensorFlow's `model.summary()` function.

4.2. Evaluation Metrics

The performance of the proposed model is evaluated using categorical cross-entropy as the loss function, which is suitable for multi-class classification problems. Classification accuracy is used as the primary evaluation metric to measure the proportion of correctly classified images during both training and testing phases.

4.3. Dataset Split

The CIFAR-10 dataset is divided into a training set of 50,000 images and a test set of 10,000 images. The training set is used for learning model parameters, while the test set is used to evaluate generalization performance.

4.4. Implementation Details

The model is implemented using the TensorFlow and Keras deep learning framework. Training is performed using the Adam optimizer due to its adaptive learning rate and efficient convergence properties. The network is trained for a limited number of epochs with a fixed batch size to balance learning efficiency and computational cost.

All experiments are conducted on normalized image data, and model performance is evaluated on both the test dataset and external images.

4.5. Other Methods

No additional optimization techniques such as data augmentation, dropout, or batch normalization were employed in the initial implementation. The model architecture was intentionally kept simple to maintain interpretability and originality, while ensuring a clear demonstration of fundamental convolutional neural network concepts.

5. Results and Discussion

5.1. Results

The proposed convolutional neural network was trained for 15 epochs on the CIFAR-10 training dataset. During training, the model showed a steady improvement in classification

accuracy and a corresponding reduction in loss values, indicating effective learning of visual features from the input images.

The final evaluation on the CIFAR-10 test dataset yielded a test accuracy of approximately **69.24%**, demonstrating that the trained model is able to generalize reasonably well to unseen data. The final test loss was observed to be approximately **0.96**, which is consistent with the complexity of the dataset and the simplicity of the chosen architecture.

In addition to standard test set evaluation, the trained model was tested using external images obtained from the internet. After appropriate preprocessing and resizing, the model was able to correctly predict the object category for the tested images, indicating satisfactory generalization performance beyond the original dataset.

5.2. Visualizations

The training and validation accuracy and loss curves across epochs provide insight into the model's learning behavior. Training accuracy increased consistently from approximately 44% in the first epoch to nearly 80% by the final epoch, while validation accuracy stabilized around 69–70%, indicating effective learning without severe overfitting.

The loss curves show a gradual decrease in training loss, with validation loss remaining relatively stable, suggesting a balanced bias–variance trade-off.

5.3. Performance Comparison

To assess the effectiveness of the proposed approach, the performance of the convolutional neural network was compared with a simpler baseline consisting of shallow learning approaches typically used for image classification. Unlike traditional machine learning models that rely on handcrafted features, the CNN automatically learned hierarchical feature representations directly from raw pixel data.

The obtained test accuracy of approximately 69% highlights the effectiveness of convolutional architectures for image recognition tasks.

5.4. Analysis

The experimental results indicate that the convolutional neural network successfully captures discriminative visual features from the CIFAR-10 dataset. The gradual improvement in accuracy across epochs suggests stable convergence during training.

Some misclassifications were observed among visually similar classes, which can be attributed to the low resolution of CIFAR-10 images and the limited depth of the network. Nevertheless, the model demonstrates robust performance given its simple design.

5.5. Improvements

Future work may explore deeper architectures, regularization techniques such as dropout or batch normalization, data augmentation, and transfer learning to further improve performance.

6. References

- Krizhevsky, A., Hinton, G. (2009). *Learning Multiple Layers of Features from Tiny Images*. CIFAR-10 Dataset. <https://www.cs.toronto.edu/~kriz/cifar.html>
- Kaggle. *CIFAR-10 Object Recognition Competition*. <https://www.kaggle.com/competitions/cifar-10>