

# Project Report: CIFAR-10 Image Classification using Deep Learning

## Introduction

This project focuses on classifying images from the CIFAR-10 dataset using deep learning techniques. The CIFAR-10 dataset consists of 60,000 32x32 color images in 10 different classes, with 6,000 images per class. The goal is to build and evaluate two types of neural networks: an Artificial Neural Network (ANN) and a Convolutional Neural Network (CNN), to determine their effectiveness in this image classification task.

## Data Loading and Preprocessing

The CIFAR-10 dataset was loaded using the `datasets.cifar10.load_data()` function from TensorFlow Keras. The dataset was split into training and testing sets, `X_train`, `y_train`, `X_test`, and `y_test`.

The shape of a single label in the training set was initially (1, ), so the training labels `y_train` were reshaped to (-1, ) to match the expected input shape for the model.

The pixel values of the images were normalized by dividing by 255. This scales the pixel values to the range [0, 1], which is a common practice for image data and helps in better model training.

## Artificial Neural Network (ANN) Model

An Artificial Neural Network (ANN) was constructed with the following architecture:

- A Flatten layer to transform the 32x32x3 images into a 1D array.
- A dense layer with 3000 units and ReLU activation.
- A dense layer with 1000 units and ReLU activation.
- An output dense layer with 10 units (for the 10 classes) and a sigmoid activation function.

The model was compiled using the Stochastic Gradient Descent (SGD) optimizer and `sparse_categorical_crossentropy` as the loss function, with 'accuracy' as the evaluation metric.

The ANN was trained for **5** epochs. The training accuracy improved over the epochs, reaching approximately **49.29%** in the final epoch.

The model was evaluated on the test set, resulting in an accuracy of approximately **49%** and a loss of **1.4357**

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.60      | 0.52   | 0.56     | 1000    |
| 1            | 0.66      | 0.49   | 0.56     | 1000    |
| 2            | 0.40      | 0.37   | 0.38     | 1000    |
| 3            | 0.37      | 0.25   | 0.30     | 1000    |
| 4            | 0.48      | 0.36   | 0.41     | 1000    |
| 5            | 0.42      | 0.36   | 0.39     | 1000    |
| 6            | 0.48      | 0.60   | 0.53     | 1000    |
| 7            | 0.58      | 0.52   | 0.55     | 1000    |
| 8            | 0.60      | 0.64   | 0.62     | 1000    |
| 9            | 0.39      | 0.76   | 0.52     | 1000    |
| accuracy     |           |        | 0.49     | 10000   |
| macro avg    | 0.50      | 0.49   | 0.48     | 10000   |
| weighted avg | 0.50      | 0.49   | 0.48     | 10000   |

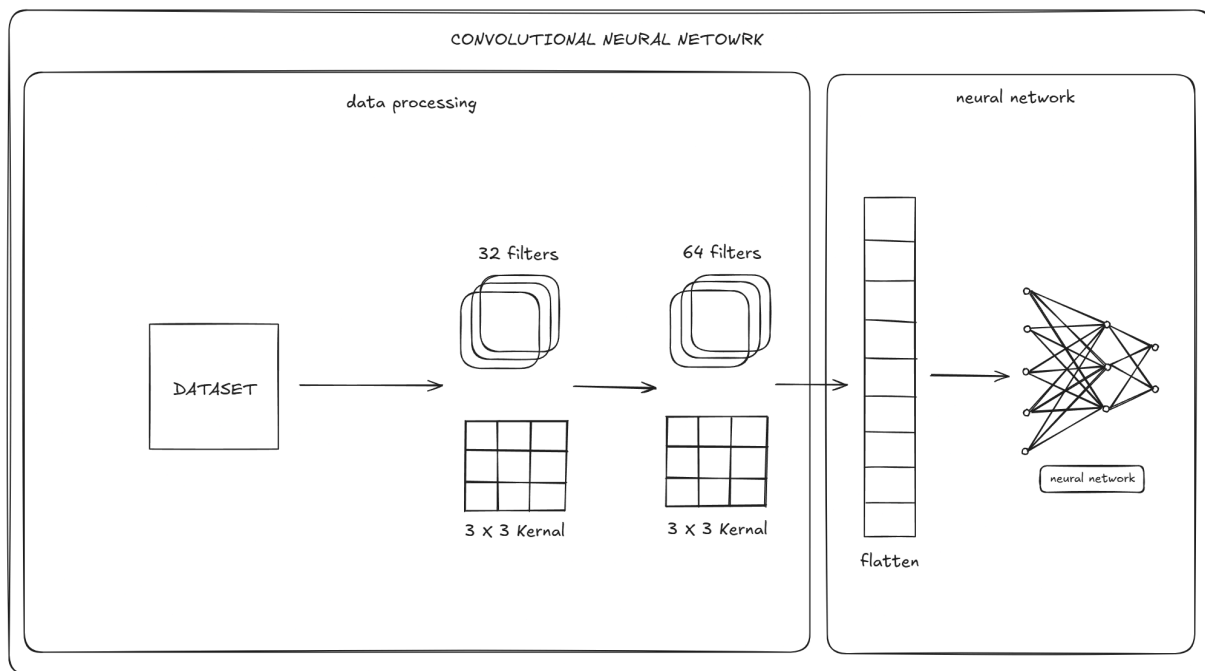
**Fig 1: classification report of ANN**

## Convolutional Neural Network (CNN) Model

A Convolutional Neural Network (CNN) was constructed with the following architecture:

- A 2D convolutional layer with 32 filters, a kernel size of (3, 3), and ReLU activation, with input shape (32, 32, 3).
- A MaxPooling2D layer with a pool size of (2, 2).
- A 2D convolutional layer with 64 filters, a kernel size of (3, 3), and ReLU activation.

- A MaxPooling2D layer with a pool size of (2, 2).
- A Flatten layer to transform the output of the convolutional layers into a 1D array.
- A dense layer with 64 units and ReLU activation.
- An output dense layer with 10 units (for the 10 classes) and a softmax activation function.



The model was compiled using the 'adam' optimizer and `sparse_categorical_crossentropy` as the loss function, with 'accuracy' as the evaluation metric.

The CNN was trained for **8** epochs. The training accuracy improved significantly over the epochs, reaching approximately **77.38%** in the final epoch.

The model was evaluated on the test set, resulting in an accuracy of approximately **69.64%** and a loss of **0.9091**. The classification report shows varying precision, recall, and f1-scores for each class, with overall macro and weighted averages around **69%**.

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.65      | 0.83   | 0.73     | 1000    |
| 1            | 0.80      | 0.80   | 0.80     | 1000    |
| 2            | 0.75      | 0.42   | 0.54     | 1000    |
| 3            | 0.50      | 0.54   | 0.52     | 1000    |
| 4            | 0.67      | 0.60   | 0.63     | 1000    |
| 5            | 0.61      | 0.62   | 0.62     | 1000    |
| 6            | 0.83      | 0.73   | 0.78     | 1000    |
| 7            | 0.69      | 0.78   | 0.73     | 1000    |
| 8            | 0.83      | 0.76   | 0.79     | 1000    |
| 9            | 0.69      | 0.86   | 0.76     | 1000    |
| accuracy     |           |        | 0.69     | 10000   |
| macro avg    | 0.70      | 0.70   | 0.69     | 10000   |
| weighted avg | 0.70      | 0.69   | 0.69     | 10000   |

Fig 2: classification report of CNN

## Comparison of Models

Comparing the performance of the **ANN** and **CNN** models on the **CIFAR-10** test dataset reveals a significant difference. The **ANN** achieved a test accuracy of approximately **49.29%**, while the **CNN** achieved a much higher test accuracy of approximately **69.64%**.

| Performance            | Artificial Neural Network | Convolutional Neural Network |
|------------------------|---------------------------|------------------------------|
| No of Epochs           | 5                         | 8                            |
| Training data Accuracy | ~49%                      | ~77%                         |
| Training Data Loss     | 1 . 4409                  | 0 . 6446                     |
| Test Data Accuracy     | ~49.5%                    | ~69.6%                       |
| Test Data Loss         | 1 . 4354                  | 0 . 9091                     |

The superior performance of the CNN is expected for image classification tasks. Convolutional layers are designed to capture spatial hierarchies and patterns in images through the use of filters and pooling, which are crucial for recognizing visual features. ANNs, on the other hand, process flattened image data, losing valuable spatial information. The results clearly demonstrate the effectiveness of CNNs in handling image data compared to standard ANNs. The classification report for the CNN further provides detailed metrics for each class, showing varying degrees of performance across the different categories

## Conclusion

In conclusion, both Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN) were implemented and evaluated for the CIFAR-10 image classification task. The results clearly show that the CNN model significantly outperformed the ANN model, achieving a test accuracy of 67.86% compared to the ANN's 44.69%. This reinforces the suitability of CNNs for image-based tasks due to their ability to effectively learn spatial features.

Future work could involve exploring more complex CNN architectures, incorporating techniques like data augmentation, batch normalization, and dropout to further improve model performance and generalization. Additionally, experimenting with different optimizers and learning rates could potentially lead to better results.

### Project Completion:

Nakka Kiran Siva Koushik