Exploring Convolutional Neural Networks (CNNs)

# Building a Basic CNN

Using TensorFlow or PyTorch, implement a basic CNN with the following specifications:

* Input Layer: Suitable for the chosen dataset
* Convolutional Layers: At least two convolutional layers with ReLU activation
* Pooling Layers: At least one max pooling layer
* Fully Connected Layer: At least one fully connected layer at the end
* Output Layer: Suitable for the classification task of the dataset

**Dataset**: Use the CIFAR 100 dataset for handwritten digit classification.

**Training**: Train the model on the CIFAR 100 dataset and document the training process, including any challenges faced and how they were addressed.

**Evaluation**: Evaluate the model on the test set and report accuracy. Also, include a confusion matrix to understand the model's performance better.

# Experimentation and Analysis

1. **Filter Analysis**: Visualize the filters and feature maps in the first convolutional layer. Discuss what features you think the network is learning at this layer.
2. **Hyperparameter Tuning**: Experiment with different hyperparameters, including the number of filters, kernel size, and learning rate. Record the impact of these changes on model accuracy and training time.
3. **Pooling Analysis**: Replace max pooling with average pooling and observe any changes in performance and model accuracy.
4. **Advanced Architecture**: Implement a more complex CNN architecture (like LeNet or a simplified version of AlexNet) and compare its performance with the basic CNN implemented earlier.