ML Lab Week 14: CNN Image Classification

Name : Nikitha P
SRN : PES2UG23CS389
SEM : 5F
DATE : 20/11/2025

1. Problem Statement

The objective of this lab was to design, train, and evaluate a Convolutional Neural Network (CNN) using PyTorch for classifying hand-gesture images into three distinct categories: **rock, paper, and scissors**.
Using the provided Jupyter Notebook template, the task involved completing the missing code, building the model, training it on a labeled dataset, and analyzing its performance.

2. Report Contents

1. **Introduction:**

The aim of this experiment was to develop a CNN capable of recognizing and categorizing hand-gesture images into Rock, Paper, or Scissors. The dataset, sourced from Kaggle, consisted of 128×128 RGB images that required preprocessing before being fed into the network.
By training a deep learning model to learn spatial patterns from the images, the goal was to achieve reliable classification accuracy on both training and unseen test samples, highlighting the suitability of CNNs for image-based tasks.

2. **Model Architecture:**

   The designed CNN contains three main convolutional stages, gradually increasing the number of channels from 3 to 16, 32, and 64. Each stage includes a 3×3 convolution (padding = 1), followed by a ReLU activation and a MaxPool2d layer with a 2×2 kernel to reduce spatial resolution.
   After feature extraction, the output feature map of size 64×16×16 is flattened and passed through a fully connected block: a linear layer with 256 units, a ReLU activation, and a Dropout layer with probability 0.3 to reduce overfitting.
   The network ends with a final linear layer producing three output scores representing the three gesture classes.

3. **Training and Performance:**

   Training was carried out using the Adam optimizer (learning rate = 0.001) and CrossEntropyLoss, which is suitable for multi-class classification problems. The model was trained for 10 epochs with batches of size 32, allowing the parameters to be updated iteratively based on the computed gradients.
   The network trained smoothly and achieved a final test accuracy of 98.17%, demonstrating that the chosen architecture and hyperparameters were well-suited to the dataset and classification task.

4. **Conclusion and Analysis:**

   The trained model delivered excellent performance, achieving over 98% accuracy on the test dataset. Although overfitting was a possible concern due to limited data variability, the inclusion of a Dropout layer helped maintain strong generalization.
   For further improvements, techniques such as data augmentation—like random rotations, flips, or intensity variations—could make the model more resilient to different hand positions. Additionally, incorporating early stopping could prevent unnecessary training once performance levels off.