ML Lab Week 14: CNN Image Classification

Name : N S Likhith Chandra
SRN : PES2UG23CS366
SEM : 5F
DATE : 19/11/2025

1. Problem Statement

The goal of this lab is to design, build, and train a Convolutional Neural Network (CNN) using PyTorch. This model must be able to accurately classify images of hand gestures into one of three categories: 'rock', 'paper', or 'scissors'.

You will be given a boilerplate Jupyter Notebook. Your task is to complete the missing code, train the model, evaluate its performance, and write a report on your findings.

2. Report Contents

1. **Introduction:**

   The primary objective of this lab was to design, implement, and train a Convolutional Neural Network (CNN) using PyTorch to accurately classify images of hand gestures into three categories: Rock, Paper, and Scissors. The project involved handling a dataset from Kaggle, preprocessing 128x128 pixel images, and building a deep learning model capable of extracting spatial features. By training this model on a labeled dataset, the goal was to achieve high classification accuracy on unseen test data, demonstrating the effectiveness of CNNs in computer vision tasks

2. **Model Architecture:**

   The architecture consists of three convolutional blocks, where the channel depth increases from 3 to 16, then 32, and finally 64. Each block utilizes a Conv2d layer with a 3x3 kernel and padding of 1, followed by a ReLU activation and a MaxPool2d layer (kernel size 2) to downsample the spatial dimensions. The classifier flattens the resulting feature map (64x16x16) and passes it through a fully connected block consisting of a linear layer (256 units), a ReLU activation, and a Dropout layer (p=0.3) to mitigate overfitting. The final output is a linear layer with 3 neurons corresponding to the three target classes.

3. **Training and Performance:**

   The model was trained using the Adam optimizer with a learning rate of 0.001 and the CrossEntropyLoss function, which is standard for multi-class classification tasks. The training process was executed for 10 epochs with a batch size of 32, allowing the model to iteratively update its weights based on the calculated gradients. The model converged successfully, achieving a Final Test Accuracy of 98.17%. This high accuracy score indicates that the chosen hyperparameters and architecture were highly effective for this specific dataset.

4. **Conclusion and Analysis:**

   The results indicate that the model performed exceptionally well, generalizing effectively to the test set with over 98% accuracy. One potential challenge faced was the risk of overfitting, the high test score suggests the Dropout layer successfully maintained generalization. To further improve the model in the future, data augmentation techniques such as random rotations or horizontal flips could be introduced to make the network more robust to varied hand orientations. Additionally, implementing early stopping could optimize training time by halting the process once accuracy stabilizes.