ML Lab Week 14: CNN Image Classification

Name: Nehal G

SRN: PES2UG23CS380

Section: F

1. Introduction

The objective of this lab was to design, build, and train a Convolutional Neural Network (CNN) using PyTorch to classify images into one of three categories: rock, paper, or scissors. The dataset consisted of over 2,000 images of hand gestures organized into separate folders for each class. The model was trained using these labeled images to learn distinguishing visual features, and its performance was evaluated using a separate test dataset. Additionally, the trained model was tested on random images to simulate a practical rock-paper-scissors prediction scenario.

2. Model Architecture

The CNN used in this lab consists of three convolutional blocks. Each block includes a convolutional layer with a kernel size of 3 and padding of 1, followed by a ReLU activation function and max pooling with a pool size of 2. The number of channels increases progressively through the layers as follows: the first convolutional layer maps 3 input channels to 16 output channels, the second layer maps 16 to 32 channels, and the third layer maps 32 to 64 channels. After passing through the three max pooling layers, the spatial resolution of the input image is reduced from 128 by 128 pixels to 16 by 16 pixels. This results in a flattened feature size of 64 multiplied by 16 multiplied by 16, which is 16,384.

The fully connected classifier block consists of a flattening operation followed by a linear layer that reduces the feature vector from 16,384 to 256 dimensions. A ReLU function is applied after this layer, followed by a dropout layer with probability 0.3 to reduce overfitting. Finally, a second linear layer maps the 256 features to 3 output units, corresponding to the three gesture classes.

3. Training and Performance

The model was trained for 10 epochs using the Adam optimizer with a learning rate of 0.001. The loss function used during training was CrossEntropyLoss, which is suitable for multi-class classification problems. A batch size of 32 was used during data loading. During training, loss decreased significantly across epochs, starting from 0.7438 in the first epoch and reducing to 0.0023 by the final epoch. After training, the model achieved a test accuracy of 98.40 percent on the validation dataset, indicating strong generalization capability.

4. Conclusion and Analysis

The trained CNN model performed very well, achieving high accuracy on the test dataset. The decreasing loss values across epochs indicate effective learning, and the high final accuracy demonstrates successful classification of image data. Testing on random images also confirmed correct predictions, and the model was successfully applied to simulate a rock-paper-scissors game using real dataset samples.

One challenge faced during implementation was determining the correct flattened size for the fully connected layer after the convolutional and pooling operations. Another minor challenge involved ensuring consistent image transformations during both training and testing phases.

To further improve the model, data augmentation techniques could be applied to make the model more robust to variations in lighting, rotation, and orientation. Additionally, experimenting with deeper network architectures or pretrained models such as ResNet may improve the accuracy further. Adjusting dropout rates and implementing learning rate scheduling could also help fine-tune performance and reduce slight overfitting observed in later epochs.