ML Lab- 14

***Name**: Maya Nithyanand Bhagath*

***SRN**: PES2UG23CS331*

Introduction

The objective of this lab was to design, build and train a convoluted neural network (CNN) using PyTorch. The goal was to accurately classify images of hand gestures into one of the three categories – rock, paper or scissor. The process involved downloading the dataset, preprocessing the images, implementing a custom CNN architecture, and evaluating the model's accuracy on an unseen test set.

Model Architecture

CNN Architecture

This model has 2 main components – the convolution layers and the fully connected classifier. The feature extractor has three sequential blocks. Each block consists of a Conv2d layer with a kernel size of 3 and padding of 1, followed by a ReLU activation function and a Max Pooling layer with a kernel size of 2 and stride of 2. The depth of the channels gradually increases. Initially it has 3 input channels, and it then eventually increases to 64 channels.

Key Parameters

- Kernel size – 3x3 with padding of 1
- No. of channels – initially 3, then increases to 16, then 32, then finally 64
- Max Pooling – uses kernel of size 2x2 and default stride of 2

Fully-Connected Classifier

The fully connected classifier flattens the output of the final convolution block into a vector of size 64x16x16. The classifier consists of a linear layer, a ReLU activation function, a dropout layer, and another linear layer.

Training and Performance

Key Hyperparameters for Training

- Optimizer – Adam
- Loss Function – CrossEntropyLoss
- Learning Rate – 0.001
- Epochs – 10
- Batch Size - 32

Test Accuracy

The final test accuracy was 98.86%.

Conclusion and Analysis

Results

The model performed very well, achieving an accuracy of 98.86% on the test set. The steady decrease in training loss indicates that the network successfully learned the features that differentiated the three hand gestures.

Challenges

Calculating the correct input size so that the Max Pooling layers could half the dimensions was a challenge.

Potential Improvements

- The data could be augmented by adding random transformations to make it more suitable for real world variations and data.
- Stopping the training before the loss value becomes very small could save computational resources.