ML LAB-14

Name: Nikitha Devaraj

SRN : PES2UG23CS388

Date: 20.11.2025

1. Introduction

The objective of this lab was to design, build, and train a Convolutional Neural Network (CNN) using the PyTorch framework. The specific task was to classify images of hand gestures into one of three categories: 'rock', 'paper', or 'scissors'[1]. The model was trained using the "Rock Paper Scissors" dataset downloaded from Kaggle, which contains over 2,000 RGB images[2]. The project involved data preprocessing, defining the neural network architecture, training the model using backpropagation, and evaluating its accuracy on a held-out test set.

2. Model Architecture

The model architecture (RPS_CNN) consists of a feature extraction block followed by a classification block[3].

Convolutional Blocks (Feature Extraction):

The network utilizes three sequential convolutional blocks to extract spatial features from the input images

- **Block 1:** Conv2d layer with a kernel size of 3 and padding of 1. This is followed by a ReLU activation function and a MaxPool2d layer (2x2 kernel) which reduces the image size to 64×64.
- **Block 2:** Conv2d layer with a kernel size of 3 and padding of 1, followed by ReLU and MaxPool2d. Image size reduces to 32×32.
- **Block 3:** Conv2d layer with a kernel size of 3 and padding of 1, followed by ReLU and MaxPool2d. The final feature map size is 16×16 with 64 channels[4].

Fully Connected Classifier:

- **Flattening:** The output from the convolutional blocks is flattened into a single vector of size 16,384.
- **Hidden Layer:** A Linear layer maps the 16,384 inputs to 256 neurons, followed by a ReLU activation.
- **Dropout:** A Dropout layer with a probability of $p=0.3$ is applied to prevent overfitting.
- **Output Layer:** A final Linear layer maps the 256 neurons to 3 output classes (Rock, Paper, Scissors)[5].

3. Training and Performance

The model was trained using the following hyperparameters[6]:

- **Optimizer:** Adam
- **Loss Function:** CrossEntropyLoss (suitable for multi-class classification)
- **Learning Rate:** 0.001
- **Epochs:** 10
- **Batch Size:** 32

Performance:

After training for 10 epochs, the model was evaluated on the unseen test dataset (20% of the total data).

- **Final Test Accuracy: [97.16]%**

4. Conclusion and Analysis

Results:

The CNN model performed successfully, achieving a high accuracy on the test set. The loss consistently decreased over the 10 epochs, indicating that the network successfully learned the features required to distinguish between the hand gestures8.

Challenges:

One challenge faced during the lab was ensuring the dimensions of the flattened layer matched the input of

the first linear layer. Calculations were required to track the image size reduction through the three MaxPool layers .Additionally, ensuring the data was correctly moved to the GPU (CUDA) was necessary for efficient training speeds.

Future Improvements:

To potentially improve the model's accuracy further, the following steps could be taken9:

1. **Data Augmentation:** Currently, the model only sees the raw images. Adding random rotations, horizontal flips, or color jittering to the transform pipeline would help the model generalize better to real-world variations.
2. **Transfer Learning:** Instead of training from scratch, we could use a pre-trained model like ResNet18 or VGG16, freeze the early layers, and fine-tune the classifier. This often yields higher accuracy with fewer epochs.