ML LAB 14

Name: Mrigank Jain

Semester: 5

Section: F

SRN: PES2UG23CS352

Introduction:

The objective of this laboratory exercise was to design, implement, and train a Convolutional Neural Network (CNN) using the PyTorch framework. The specific task was to build a model capable of image classification , accurately distinguishing between three distinct hand gestures: 'rock', 'paper', and 'scissors'. The project involved completing a boilerplate notebook, preprocessing the "Rock Paper Scissors" dataset, defining a custom CNN architecture, and training the model to evaluate its final performance on unseen test data.

Model Architecture:

The classification task was handled by a custom CNN model named RPS_CNN. The network consists of a feature extraction block (convolutional layers) and a classifier block (fully-connected layers).

Convolutional Block (self.conv_block)

The feature extractor is composed of three sequential convolutional layers, each followed by a ReLU activation and a MaxPool2d(2) layer. The input images were preprocessed to 128 x 128 pixels, and the spatial dimensions are reduced by a factor of 2 in each MaxPool layer.

- **Block 1:** Conv2d (3 input channels, 16 output channels, kernel_size=3, padding=1), ReLU, MaxPool2d(2).
- **Block 2:** Conv2d (16 input channels, 32 output channels, kernel_size=3, padding=1), ReLU, MaxPool2d(2).
- **Block 3:** Conv2d (32 input channels, 64 output channels, kernel_size=3, padding=1), ReLU, MaxPool2d(2).

Fully-Connected Classifier (self.fc)

The output of the final convolutional block is passed to the classifier. After the three MaxPool(2) layers, the spatial size is reduced to 16 x 16, resulting in 64 feature maps.

- **Flatten Layer:** The feature maps are flattened, yielding an input size of 16384 features (64 x 16 x 16).
- **Layer 1:** A linear layer with weights mapping 16384 inputs to 256 outputs, followed by a ReLU activation.
- **Regularization:** A **Dropout** layer with a probability p=0.3 was applied to prevent overfitting.

- **Output Layer:** The final linear layer maps 256 inputs to 3 outputs, corresponding to the three classes.

Training and Performance:

Hyperparameters

The model was trained using the following key hyperparameters:

- **Optimizer:** Adam
- **Loss Function (Criterion):** CrossEntropyLoss.
- **Learning Rate (lr):** 0.001.
- **Number of Epochs:** 10.
- **Batch Size:** 32.

Test Accuracy

The model was evaluated on the 20% unseen test set.

The final test accuracy is 99.54%

Conclusion and Analysis:

The CNN model demonstrated excellent performance on the image classification task, achieving a final test accuracy of 99.54%. The architecture, which utilized three convolutional blocks for feature extraction and a fully-connected layer with Dropout for classification, proved highly effective in learning the distinct features of the hand gestures.

Challenges:

A key challenge encountered during the initial setup was ensuring the data loading pipeline was correctly configured, specifically by correcting the hardcoded source path (src_root) to match the actual download location returned by kagglehub.

Suggested Improvements:

1. **Data Augmentation:** To improve the model's generalization capabilities, the input transforms should be expanded to include more robust data augmentation techniques such as random rotation, scaling, and shear transformations.

2. **Transfer Learning:** A substantial performance increase could likely be achieved by using a state-of-the-art pre-trained model as a feature extractor. This technique would allow the model to leverage features learned from millions of general images, providing a much stronger base for classification.