Week 14 Lab Report: CNN Image Classification

Name	SRN	Section
Mohammed Ehan Sheikh	PES2UG23CS345	F

Introduction

The objective of this lab is to design, build, and train a Convolutional Neural Network (CNN) using PyTorch to classify images of hand gestures into one of three categories: 'rock', 'paper', or 'scissors'. The dataset used contains over 2,000 images organized into three folders, and the model is trained to achieve high accuracy in recognizing these gestures.

Model Architecture

The CNN model, named RPS_CNN, consists of a convolutional block followed by a fully-connected classifier.

Convolutional Block

- First layer: Conv2d with 3 input channels (RGB), 16 output channels, kernel size 3x3, padding 1, followed by ReLU activation and MaxPool2d with kernel size 2.
- Second layer: Conv2d with 16 input channels, 32 output channels, kernel size 3x3, padding 1, followed by ReLU activation and MaxPool2d with kernel size 2.
- Third layer: Conv2d with 32 input channels, 64 output channels, kernel size 3x3, padding 1, followed by ReLU activation and MaxPool2d with kernel size 2.

Fully-Connected Classifier

- Flatten layer to convert the feature maps into a 1D vector.
- Linear layer with input size 64*16*16 (16384) and output size 256, followed by ReLU activation.
- Dropout layer with probability 0.3 to prevent overfitting.
- Final Linear layer with 256 inputs and 3 outputs (one for each class).

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 was CrossEntropyLoss for multi-class classification.

The training loss decreased progressively: starting from 0.5923 in epoch 1 to 0.0046 in epoch 10, indicating good convergence.

The final test accuracy achieved was 97.26%, evaluated on 438 test images out of a total of 2188 images (80% training, 20% testing split).

Conclusion and Analysis

The model performed well, achieving a high test accuracy of 97.26%, which demonstrates effective learning of the features distinguishing the three hand gestures.

No major challenges were faced during the implementation, as the boilerplate code was comprehensive and the dataset was straightforward to work with. The paths were initially set for a different environment (Kaggle/Colab), but were easily adjusted to run locally.

To potentially improve the model's accuracy further, one could: - Increase the number of epochs or use early stopping to prevent overfitting. - Experiment with data augmentation techniques, such as random rotations or flips, to increase the diversity of the training data. - Try deeper architectures or different kernel sizes in the convolutional layers.