ML LAB-14 Report

NAME:NIKHIL GARUDA

SRN:PES2UG23CS195

SECTION:C

Report Contents

1. Introduction

The objective of this lab was to build, train, and evaluate a Convolutional Neural Network (CNN) for image classification using the Rock–Paper–Scissors dataset. The goal was to understand how convolutional layers, pooling, and fully connected layers work together for visual pattern recognition. The experiment involved implementing data transformations, creating a CNN model from scratch, training it on labeled images, and testing its performance to classify gestures correctly.

2. Model Architecture

The CNN architecture implemented (SimpleRPSCNN) consists of four convolutional blocks, each followed by Batch Normalization, ReLU activation, and Max Pooling. The convolutional layers use 3×3 kernels with padding=1 and progressively increase channels from 3 → 32 → 64 → 128 → 256. After the final convolutional layer, an Adaptive Average Pooling layer reduces the feature map to a fixed 4×4 size.

The fully connected classifier includes a Dropout layer (0.4) → Linear layer (4096 → 512) → ReLU → Dropout(0.4) → Linear(512 → 3). This design balances feature extraction and generalization while maintaining computational efficiency.

3. Training and Performance

- **Optimizer:** Adam
- **Loss Function:** CrossEntropyLoss
- **Learning Rate:** 0.001
- **Batch Size:** 32
- **Epochs:** 15
- **Scheduler:** ReduceLROnPlateau (monitored validation accuracy)
- **Final Test Accuracy: 100%**

The model achieved perfect test accuracy, successfully classifying all images in the test dataset. This indicates strong learning of class-specific visual features.

4. Conclusion and Analysis

The CNN performed exceptionally well, reaching 100% test accuracy. This result shows that the model effectively distinguished between rock, paper, and scissors gestures in the given dataset. However, such perfect performance may suggest that the dataset is relatively simple or that training, validation, and test sets have similar samples, possibly leading to overfitting.

Challenges faced:

- A minor compatibility issue with the PyTorch learning rate scheduler (resolved by removing the verbose=True argument).
- Occasional worker errors in Colab's DataLoader, fixed by setting num_workers=0.

Future improvements:

1. Use transfer learning with a pretrained CNN like ResNet18 or MobileNetV2 for better generalization.
2. Introduce stronger data augmentation (cropping, color jitter, blur) and collect more diverse images to improve robustness and prevent overfitting.

Final Outcome:

The experiment successfully demonstrated CNN-based image classification, with all implementation goals achieved and a test accuracy of 100%.