ML Lab Week 14: CNN Image Classification

Name:Bobba Koushik
Srn:pes2ug23cs133
Sec:c

1. Introduction

The objective of this lab was to design, implement, and train a Convolutional Neural Network (CNN) capable of classifying images of hand gestures into the categories *rock*, *paper*, and *scissors*. Using PyTorch, the dataset was preprocessed, split into training and testing sets, and fed into a custom-built CNN model. The trained model was then evaluated for accuracy and tested on new images to verify its performance.

Convolutional Layers

The CNN consisted of **three convolutional blocks**, each containing:

Conv2d layer with kernel size **3×3** and padding **1**

ReLU activation

MaxPooling layer with pool size **2×2**

The architecture details:

Block 1

Input channels: **3** (RGB)

Output channels: **16**

Kernel size: **3×3**, padding=1

MaxPool reduces spatial size from 128×128 → 64×64

Block 2

Input channels: **16**

Output channels: **32**

MaxPool reduces from 64×64 → 32×32

Block 3:

Input channels: **32**

Output channels: **64**

MaxPool reduces from 32×32 → 16×16

Fully Connected Classifier

After flattening the output from the convolutional layers (size: **64 × 16 × 16**):

Linear layer: 16384 → 256

ReLU activation

Dropout layer: p = 0.3 to reduce overfitting

Final Linear layer: 256 → 3 (three gesture classes)

This classifier block maps the extracted image features into the final gesture categories.

3. Training and Performance

Training Hyperparameters

The model was trained using the following settings:

Optimizer: Adam

Loss function: CrossEntropyLoss

Learning rate: 0.001

Epochs: 10

Batch size: 32

Train/Test split: 80% training, 20% testing

These parameters allowed the model to converge efficiently while avoiding overfitting.

Final Test Accuracy

After completing training, the model achieved a **test accuracy of approximately 97.72%**

Conclusion and Analysis

The CNN performed well on the Rock-Paper-Scissors dataset and achieved high accuracy, demonstrating that the architecture was effective for image classification. The use of three convolutional blocks with max pooling helped extract strong spatial features, while the fully connected layers successfully mapped these features to class labels.

Challenges Faced

Ensuring correct dataset paths and folder structure

Choosing the right image transformations (resize + normalization)

Understanding how dimensions shrink after each pooling layer

GPU/CPU runtime differences

Possible Improvements

Data Augmentation:
Adding random flips, rotations, or color jitter could help the model generalize better.

Deeper Model / Batch Normalization:
Increasing the number of convolution layers or adding BatchNorm can improve accuracy.

More Epochs:
Training for 20–25 epochs often results in slightly higher accuracy