

ML LAB-5

NAME: Nilay Srivastava

SRN: PES2UG23CS390

SECTION: F

I. INTRODUCTION

The objective of this lab was to design, train and evaluate a CNN to classify images of hand gestures into three categories: rock, paper, and scissors. Using PyTorch, we built a complete image classification pipeline including image preprocessing, model architecture definition, training loops, evaluation techniques and prediction on single images. The goal was to understand how CNNs learn spatial information and achieve high accuracy on real-world image datasets.

II. MODEL ARCHITECTURE

The CNN designed in this experiment consists of three convolutional blocks, each followed by ReLU activation and MaxPooling. The network progressively extracts hierarchical spatial features and reduces image resolution before passing the flattened representation into a fully connected classifier.

Convolution Blocks:

1. (conv_block): Sequential(
 (0): Conv2d(3, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
2. (3): Conv2d(16, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
3. (6): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))

```

RPS_CNN(
    (conv_block): Sequential(
        (0): Conv2d(3, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (1): ReLU()
        (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
        (3): Conv2d(16, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (4): ReLU()
        (5): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
        (6): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
        (7): ReLU()
        (8): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    )
    (fc): Sequential(
        (0): Flatten(start_dim=1, end_dim=-1)
        (1): Linear(in_features=16384, out_features=256, bias=True)
        (2): ReLU()
        (3): Dropout(p=0.3, inplace=False)
        (4): Linear(in_features=256, out_features=3, bias=True)
    )
)

```

Each MaxPool operation reduces the image size by half:

128 -> 64 -> 32 -> 16, resulting in a final feature map size of $64 \times 16 \times 16 = 16384$.

Fully Connected Classifier:

The classifier converts extracted features into class probabilities:

- Flatten()
- Linear(16384->256)
- ReLU()
- Dropout(p=0.3)
- Linear(256->3)

The dropout layer reduces overfitting and improves generalization.

III. TRAINING & PERFORMANCE

Key Hyperparameters Used:

- **Optimizer:** Adam
- **Loss Function:** CrossEntropyLoss
- **Learning rate:** 0.001
- **Number of Epochs:** 10

Final Test Accuracy: 98.63%

Test Accuracy: 98.63%

IV. CONCLUSION & ANALYSIS

The CNN performed extremely well, reaching a 98.63% test accuracy. The architecture effectively captured visual patterns in hand gesture images, and the combination of convolution, pooling, and dropout contributed to stable learning and generalization.

Challenges Faced:

- Ensuring transforms (resize, normalization) were correctly applied.
- Managing the image paths and dataset structure.
- Tuning the architecture to avoid overfitting.

Potential Improvements:

1. Data Augmentation: Adding transforms such as rotations, flips, or color jitter can improve robustness.

2. More Regularization: Adding Batch Normalization or increasing dropout can stabilize training further.

3. Deeper Model: Adding more convolutional layers or using architectures like ResNet18 (transfer learning) can push accuracy even higher.